Imperial College London
Department of Computing

Scalable Performance Analysis of Massively Parallel Stochastic Systems

Richard Alexander Hayden

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in the Department of Computing of Imperial College London
April 2011

Abstract

The accurate performance analysis of large-scale computer and communication systems is directly inhibited by an exponential growth in the state-space of the underlying Markovian performance model. This is particularly true when considering massively-parallel architectures such as cloud or grid computing infrastructures. Nevertheless, an ability to extract quantitative performance measures such as passage-time distributions from performance models of these systems is critical for providers of these services. Indeed, without such an ability, they remain unable to offer realistic end-to-end service level agreements (SLAs) which they can have any confidence of honouring. Additionally, this must be possible in a short enough period of time to allow many different parameter combinations in a complex system to be tested. If we can achieve this rapid performance analysis goal, it will enable service providers and engineers to determine the cost-optimal behaviour which satisfies the SLAs.

In this thesis, we develop a scalable performance analysis framework for the grouped PEPA stochastic process algebra. Our approach is based on the approximation of key model quantities such as means and variances by tractable systems of ordinary differential equations (ODEs). Crucially, the size of these systems of ODEs is *independent* of the number of interacting entities within the model, making these analysis techniques extremely scalable. The reliability of our approach is directly supported by convergence results and, in some cases, explicit error bounds.

We focus on extracting *passage-time* measures from performance models since these are very commonly the language in which a service level agreement is phrased. We design scalable analysis techniques which can handle passages defined both in terms of *entire component populations* as well as *individual* or *tagged* members of a large population.

A precise and straightforward *specification* of a passage-time service level agreement is as important to the performance engineering process as its evaluation. This is especially true of large and complex models of industrial-scale systems. To address this, we introduce the *unified stochastic probe* framework. Unified stochastic probes are used to generate a model augmentation which exposes explicitly the SLA measure of interest to the analysis toolkit. In this thesis, we deploy these probes to define many detailed and derived performance measures that can be automatically and directly analysed using rapid ODE techniques. In this way, we tackle applicable problems at many levels of the performance engineering process: from specification and model representation to efficient and scalable analysis.

Acknowledgements

Firstly, I would like to offer my sincerest thanks and appreciation to my supervisor Dr. Jeremy Bradley for his continued expert guidance, support and advice, on all aspects of the Ph. D. journey. Further deep gratitude is due to Jeremy for giving me the initial opportunity to explore this exciting area of research all those years ago at the beginning of my final undergraduate year. I would also like to take this opportunity to apologise to, and to thank Jeremy's wife, Helen, for any late nights where Jeremy was kept back at the office, putting the finishing touches to a paper.

To the rest of the AESOP research group[1] I also owe a mighty debt of gratitude for providing such a positive and friendly working environment. In particular, my second supervisor Dr. Will Knottenbelt, the head of the group Prof. Pete Harrison, my office mates, colleagues and the 'coffee club' Chris Günther, Dr. Uli Harder, Gareth Jones, Anton Stefanek and Iryna Tsimashenka have all helped make the past few years not only intellectually stimulating, but also, great fun! I am particularly grateful to Anton for this staunch work in developing the Grouped PEPA Analyser (GPA) tool, which has both aided in the dissemination of the research presented in this thesis, and provided me with a straightforward means by which to generate the numerical results! Drs. Ashok Argent-Katwala, Nicholas Dingle, Tamas Suto and Dave Thornley, who have since managed to achieve escape velocity, must also be thanked for their part in my AESOP experience. The Computing Support Group at Imperial (CSG), in particular Lloyd Kamara, Dr. David McBride, Duncan White, and the since departed Matt Johnson, are an absolute pleasure to work with and their remarkable expertise has made the journey a much smoother one. Similarly, I thank Ann Halford for always being on hand to remove administrative headaches, and also intellectual headaches, through the reliable prescription of caffeine.

Prof. Jane Hillston and her group at the University of Edinburgh[2] also deserve many thanks for encouraging and supporting my research, for hosting many exciting PASTA workshops, and, of course, for developing the Performance Evaluation Process Algebra (PEPA) formalism, and the original, ground-breaking fluid-analysis technique which motivates the work considered in this thesis. In addition, it is always a pleasure to collaborate with Dr. Nigel Thomas of Newcastle University on the AMPS project. From my external examiner Dr. Gethin Norman of the University of Glasgow, I could not have asked for a more diligent and careful assessment of my work, for which I am extremely grateful.

For keeping me sane, particular during the seemingly non-finite writing-up period, I owe an equally non-finite set of thanks to my variously and by no means mutually-exclusively, housemates, former housemates and/or friends Inji Darwish, Heleri Hirsnik, John Fricis Ievins, Omar Katamish, Dhruve Patel, Ashley Taheri-Kadkhoda and Alfredo Carpineti, who deserves particular thanks for being my gym partner for the last few years and thus forcing me to get out of bed at the ungodly (for a Ph. D. student) hour of 8am.

Most importantly, none of this would have ever been possible without the continued and unwavering support of my parents and grandparents, my siblings and James, my rock. In these cases, words simply are not sufficient.

[1] http://aesop.doc.ic.ac.uk

[2] http://www.dcs.ed.ac.uk/pepa

Contents

Abstract **i**

Acknowledgements **iii**

1 Introduction **1**

1.1 Motivation and objectives 1

1.2 Thesis summary 4

1.3 Applications and tool support 6

1.4 Publications 6

2 Background theory **11**

2.1 Stochastic processes 11

2.2 Solving continuous-time Markov chains 13

2.2.1 Naïve numerical solution of the underlying linear system 14

2.2.2 Uniformisation 15

2.2.3 State-space aggregation and lumpability 15

2.2.4 Model representation techniques for more efficient solution 17

2.2.5 Product-form solution and mean value analysis 18

2.3 PEPA 18

2.3.1 Introduction 18

2.3.2 Syntax 19

2.3.3 Semantics 19

2.3.4 Example 21

2.4 Related work 21
2.4.1 *Fluid-flow* analysis of PEPA models 23
2.4.2 Differential equation-based approaches for sCCP and π-calculus 25
2.4.3 Mean-field models for computer and communication systems 26
2.4.4 Heavy-traffic approximations for queueing networks 28
2.4.5 Fluid/hybrid stochastic Petri nets and fluid queues 28
2.4.6 Continuous stochastic Petri nets 29

3 Moment approximations by differential equations for grouped PEPA 30

3.1 Introduction 30
3.2 Grouped PEPA models 31
3.2.1 Syntax 32
3.2.2 Semantics 34
3.2.3 Running example: client–server model 35
3.2.4 Aggregation of GPEPA models 35
3.2.5 Evolution rates of GPEPA models 37
3.3 Approximation of first moments 39
3.3.1 Nature of the first-moment approximation 41
3.3.2 First-moment approximation for running example 44
3.4 Approximation of higher moments 44
3.4.1 Joint evolution rates of GPEPA models 46
3.4.2 Action of the infinitesimal generator on higher-order moments 47
3.4.3 Derivation of the approximating ODEs 48
3.4.4 Standard-deviation approximation for running example 50
3.5 Tractability of moment approximations 55
3.6 Conclusion 55

4 Differential-equation approximations of passage times for grouped PEPA 57

4.1 Introduction . . . 57

4.2 Passage time as time to extinction . . . 58

4.3 Global passage times . . . 59

4.3.1 Approximation by point mass . . . 60

4.3.2 Upper and lower CDF approximations . . . 60

4.4 Individual passage times . . . 65

4.4.1 Steady-state individual passage times . . . 67

4.4.2 Transient individual passage times . . . 70

4.4.3 Conditioning on transition instants . . . 73

4.5 Conclusion . . . 74

5 Unified stochastic probes 75

5.1 Introduction . . . 75

5.2 Immediate GPEPA . . . 77

5.2.1 Immediate PEPA . . . 77

5.2.2 Constructing *i*GPEPA models from *i*PEPA fluid components . . . 81

5.3 Unified stochastic probes . . . 82

5.3.1 Probe syntax . . . 83

5.3.2 Translating a local probe to *i*PEPA . . . 87

5.3.3 Locating a local probe . . . 91

5.3.4 Example measure specifications . . . 92

5.4 Differential equations from *i*GPEPA models . . . 94

5.4.1 Action-counting ODEs from *i*GPEPA models . . . 95

5.5 Differential-equation approximation of probe-specified individual passage times . 98

5.5.1 Steady-state individual passage times . . . 99

5.5.2 Transient individual passage times . . . 102

5.6 Differential-equation approximation of probe-specified global passage times . . . 104

5.7 Worked example: massively-distributed wireless network . . . 107

5.7.1 Example measurements and results . . . 109
5.8 Conclusion . . . 113

6 Theoretical convergence of the first-order approximation 115

6.1 Introduction . . . 115
6.1.1 Structurally-equivalent *i*GPEPA models . . . 116
6.2 Transient bounds and convergence . . . 117
6.3 Steady-state convergence . . . 120
6.3.1 Asymptotic stability on the reachable subset and the reduced ODE . . . 121
6.3.2 Verification of global asymptotic stability of the reduced ODE . . . 124
6.3.3 Steady-state convergence theorem . . . 127
6.4 Convergence of passage-time approximations . . . 132
6.4.1 Global passage times . . . 132
6.4.2 Individual passage times . . . 135
6.5 Conclusion . . . 139

7 Conclusion 140

7.1 Summary of achievements . . . 140
7.2 Applications . . . 142
7.3 Future work . . . 148
7.3.1 Accuracy of the ODE-based analysis approach . . . 148
7.3.2 Capturing generally-timed transitions with DDEs . . . 149
7.3.3 More efficient global optimisation . . . 149
7.3.4 Multi-stage synchronisation in massively-parallel systems . . . 149

Bibliography 152

A Chapter 3 176

A.1 Lipschitz continuity of first-moment ODEs . . . 176
A.2 Systems of ODEs . . . 179
A.2.1 Second-moment ODEs for $\boldsymbol{CS}(n, m)$ . . . 179

B Chapter 4 **181**

B.1 Systems of ODEs . . . 181

B.1.1 First-moment ODEs for $\boldsymbol{CS}'(n,m)$ . . . 181

B.1.2 First-moment ODEs for $\boldsymbol{CS}_2{}'(n,m)$ . . . 181

C Chapter 5 **183**

C.1 Immediate PEPA . . . 183

C.1.1 Well-behaved iPEPA components . . . 183

C.2 Translation of non-standard regular expressions to DFAs . . . 184

C.2.1 Both $(R_l^1; R_l^2)$ . . . 184

C.2.2 Reset (R_l^1 / R_l^2) . . . 184

C.2.3 Fail $(R_l^1 \varnothing R_l^2)$ . . . 185

D Chapter 6 **186**

D.1 Proofs . . . 186

D.1.1 Proof of Theorem 6.2.1 . . . 186

D.1.2 Proof of Theorem 6.2.2 . . . 189

D.1.3 Proof of Corollary 6.2.3 . . . 191

D.1.4 Proof of Theorem 6.3.1 . . . 192

List of Figures

2.1 Simple labelled transition system which can be interpreted as a CTMC with state space $\{1, 2, 3, 4, 5, 6\}$ and transition rate between states equal to the value labelling the corresponding arc. 13

2.2 Solutions of the Chapman–Kolmogorov forward equations for each state corresponding to the CTMC given in Figure 2.1. Initial state is 1 and rates are: $r_{t_a} = 1.0$, $r_{t_b} = 0.4$, $r_r = 4.0$. 13

2.3 Structured operational semantics of PEPA 22

2.4 PEPA description of the processor–resource model $\boldsymbol{PR}(n, m)$. 22

2.5 ODE solutions for the PEPA model $\boldsymbol{PR}(100, 40)$. Rates: $r_{t_a} = 1.0$, $r_{t_b} = 0.4$, $r_r = 4.0$. 24

3.1 GPEPA description of the simple client–server model $\boldsymbol{CS}(n, m)$. 36

3.2 Accuracy of the approximation: $\mathbb{E}[\min(N_{\boldsymbol{C}}(t), N_{\boldsymbol{S}}(t))] \approx \min(\mathbb{E}[N_{\boldsymbol{C}}(t)], \mathbb{E}[N_{\boldsymbol{S}}(t)])$ for the client–server model $\boldsymbol{CS}(n, m)$. 44

3.3 Comparison of component count expectations as computed by ODEs (solid lines) and stochastic simulation (dashed lines) for the model $\boldsymbol{CS}(100, 60)$. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 1.0$, $r_{rst} = 1.0$. 45

3.4 Parametric phase plot of ODE expectation approximation (absolute) error against switch point distance for the ***Client*** component count in the model $\boldsymbol{CS}(100, 60)$. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 1.0$, $r_{rst} = 1.0$. 45

3.5 Comparison of component-count standard deviations as computed by ODEs (solid lines) and stochastic simulation (dashed lines) for the model $\boldsymbol{CS}(100, 60)$. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 2.0$, $r_{rst} = 1.0$. 53

3.6 Comparison of component-count skewness computed by ODEs (solid lines) and stochastic simulation (dashed lines) for the model $\boldsymbol{CS}(100, 60)$. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 2.0$, $r_{rst} = 1.0$. 54

3.7 Comparison of component-count kurtosis computed by ODEs (solid lines) and stochastic simulation (dashed lines) for the model $\boldsymbol{CS}(100, 60)$. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 2.0$, $r_{rst} = 1.0$. 54

4.1 New definition of the ***Client*** fluid component in the model $\boldsymbol{CS}'(n,m)$. 59

4.2 Passage-time CDFs for half of the clients to complete their first cycle computed by stochastic simulation compared with the ODE point-mass approximation (the dashed vertical line) for the model $\boldsymbol{CS}(2n,m)$. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.5$, $r_b = 0.05$, $r_d = 2.0$, $r_{rst} = 1.0$. 61

4.3 Passage-time CDFs for varying proportions of the clients to complete their first cycle computed by stochastic simulation compared with the ODE-derived upper and lower approximation from Markov's inequality for the model $\boldsymbol{CS}(20,12)$. The dashed line represents the actual theoretical bound, again computed by stochastic simulation. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.5$, $r_b = 0.05$, $r_d = 2.0$, $r_{rst} = 1.0$. 62

4.4 Passage-time CDFs for half of the clients to complete their first cycle computed by stochastic simulation compared with the ODE-derived upper and lower approximation from Chebyshev's inequality for the model $\boldsymbol{CS}(2n,m)$. The dashed line represents the actual theoretical bound, again computed by stochastic simulation. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.5$, $r_b = 0.05$, $r_d = 2.0$, $r_{rst} = 1.0$. 65

4.5 Passage-time CDFs for a single client to complete its first cycle computed by stochastic simulation compared with the ODE-derived approximation for the model $\boldsymbol{CS}(n,m)$. Rates: $r_r = 1.0$, $r_{tmt} = 0.1$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 0.7$, $r_{rst} = 0.5$. 66

4.6 Passage-time CDFs for a single client to complete a cycle in the steady-state regime computed by stochastic simulation compared with the ODE-derived approximation for the model $\boldsymbol{CS}(n,m)$. Rates: $r_r = 1.0$, $r_{tmt} = 0.1$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 0.7$, $r_{rst} = 0.5$. 69

4.7 New definition of the ***Client*** fluid component in the model $\boldsymbol{CS}_2'(n,m)$. 70

4.8 Passage-time CDFs for a single client to complete its second cycle computed by stochastic simulation compared with the ODE-derived approximation for the model $\boldsymbol{CS}(n,m)$. Rates: $r_r = 1.0$, $r_{tmt} = 0.1$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 0.7$, $r_{rst} = 0.5$. 73

5.1 Operational semantics for immediate PEPA. 79

5.2 Translation of the local probe $(fetch \mid fail), (fetch[3] \backslash fail)$ to an iPEPA component with $\mathcal{A}(P) = \{fetch, fail\}$. 88

5.3 Translation of the local probe $(fetch \mid fail)$: **begin**, $(fetch[3] \backslash fail)$: **end** to an iPEPA component with $\mathcal{A}(P) = \{fetch, fail\}$. 89

5.4 Translation of the local probe $(fetch \mid fail)$: **begin**, $(fetch[3] \backslash fail)$: **end** to an iPEPA component with $\mathcal{A}(P) = \{fetch, fail, initialise\}$. 90

5.5 Modified ***Client*** component for the computation of the steady-state individual passage-time query of Section 4.4.1. 100

5.6 Modified ***Client*** component for the computation of the transient individual passage-time query of Section 4.4.2. 103

5.7 Wireless client fluid component in the distributed wireless network model. 107

5.8 Battery fluid component in the distributed wireless network model. 108

5.9 Network channel fluid component in the distributed wireless network model. . . 108

5.10 Translation of the local probe in the measure specification of Equation 5.24 to an *i*PEPA component. 110

5.11 Passage-time CDFs for the steady-state individual passage time specified by Equation 5.24 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 $r_{timeout} = 0.2$; and weights: $\omega_{init} = 0.07$ and $\omega_{tfr} = 0.15$. 111

5.12 Translation of the local probe in the measure specification of Equation 5.27 to an *i*PEPA component. 112

5.13 Passage-time CDFs for the transient individual passage time specified by Equation 5.27 computed by stochastic simulation 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 $r_{timeout} = 0.2$; and weights: $\omega_{init} = 0.07$ and $\omega_{tfr} = 0.15$. 112

5.14 Passage-time CDFs for the global passage time specified by Equation 5.29 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 $r_{timeout} = 0.2$; and weights: $\omega_{init} = 0.07$ and $\omega_{tfr} = 0.15$. . . . 114

6.1 ϵ-tubes enclosing the rescaled ***Client*** component count around its ODE approximation for a single trace with probability at least $1-p$ in the model $\boldsymbol{CS}(5n, 3n)$ as computed using Equation 6.1 of Theorem 6.2.1. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 1.0$, $r_{rst} = 1.0$. 118

6.2 ϵ-tubes enclosing the rescaled ***Client*** component count around its ODE approximation for a single trace with probability at least $1-p$ in the model $\boldsymbol{CS}(5n, 3n)$ as computed using Equation 6.2 of Theorem 6.2.2. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 1.0$, $r_{rst} = 1.0$. 119

6.3 Comparison of ***Client*** component count simulation results with ODE approximation for the model $\boldsymbol{CS}(5n, 3n)$. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 1.0$, $r_{rst} = 1.0$. 120

6.4 *i*GPEPA description of the model $\boldsymbol{Osc}(n)$. 127

6.5 Solutions to the rescaled ODEs for the *i*GPEPA model $\boldsymbol{Osc}(n)$ given in Figure 6.4. Rates: $r_r = 1.0$, $r_q = 1.0$, $r_s = 3.0$, $r_z = 2.0$. 127

6.6 Comparison of ***Client*** component count steady-state CDF (computed by simulation) with ODE fixed-point approximation for the model $\boldsymbol{CS}(2n, n)$. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.2$, $r_b = 0.1$, $r_d = 1.0$, $r_{rst} = 1.0$. 131

6.7 Comparison of $\boldsymbol{P}_1$ component count steady-state CDF (computed by simulation) with the ODE Birkhoff centre (the solid red line on the x-axis) for the model $\boldsymbol{Osc}(n)$. Rates: $r_r = 1.0$, $r_q = 1.0$, $r_s = 3.0$, $r_z = 2.0$. 131

7.1 Fluid components $\boldsymbol{Cust^L}$ and $\boldsymbol{Cust^H}$. 144

7.2 Fluid components $\boldsymbol{Serv^L}$ and $\boldsymbol{Serv^H}$. 144

7.3 Translation of the local probe in the measure specifications of Equations 7.1 and 7.2 and its absorbing version to *i*PEPA components. 146

7.4 ODE approximations of the passage-time CDFs for the steady-state individual passage times specified by Equations 7.1 and 7.2. The dashed lines correspond to the respective SLAs, in this case, 8 seconds 95% of the time for the high-priority customers, and 40 seconds 85% of the time for the low-priority customers. Rates: $r_{request} = 2.0, r_t^H = 1.0, r_{preempt} = 3.0, r_{proc}^L = 1.0, r_{proc}^H = 2.0, r_{break} = 1.0, r_t^B = 1.0, r_{end} = 2.0, r_{switch}^{LH} = 1.0, r_{switch}^{HL} = 1.8, r_{think} = 3.0$. Initial component counts: $n_C^L = 100$, $n_C^H = 30$, $n_S^L = 20$, $n_S^H = 20$. 146

7.5 System configurations satisfying SLAs (8 seconds 95% of the time for the high-priority customers, and 40 seconds 80% of the time for the low-priority customers) over the parameter space of $n_S^L + n_S^H$ and r_t^H. The circled configuration is that which minimises the number of required services but still satisfies both SLAs. Rates: $r_{request} = 2.0, r_{preempt} = 3.0, r_{proc}^L = 1.0, r_{proc}^H = 2.0, r_{break} = 1.0, r_t^B = 1.0, r_{end} = 2.0, r_{switch}^{LH} = 1.0, r_{switch}^{HL} = 1.8, r_{think} = 3.0$. Initial component counts: $n_C^L = 100$, $n_C^H = 30$. 147

7.6 *i*GPEPA description of the simple client–server model $\boldsymbol{CS}(n, m)$. 150

7.7 *i*GPEPA description of client–server model with two classes of client and server $\boldsymbol{CS}^*(n, o, m, p)$. 150

7.8 *i*GPEPA description of client and server fluid components with channels. 151

Chapter 1

Introduction

1.1 Motivation and objectives

Accurate performance modelling at the system design stage has never been more important than in a technological age dominated by large and complex computer and communication networks. As our reliance on such systems increases, it is paramount not only that they function correctly, but that they do so in a timely manner. The field of performance analysis aims to make probabilistic guarantees about the timed behaviour of complex systems using rigorous construction and formal analysis of mathematical models. The aim is to predict system characteristics such as response time, reliability and availability.

One example of a massively parallel and complex computer system is a commercial cloud computing infrastructure such as *Amazon's Elastic Compute Cloud (EC2)* [1], the *Google App Engine* [3] or *GoGrid* [2]. Each of these allow on-demand and rapid access to a vast and often diverse array of computer hardware, usually through a virtualised interface. This is one industry where an ability to perform scalable performance engineering could have a massive impact in allowing service providers to offer tailored quality of service guarantees to their customers.

Unfortunately, the systems are so large and the interdependencies so complex that producing formal and tractable predictive models remains a real challenge. This directly inhibits the ability of these service providers to offer detailed end-to-end service guarantees which they can be confident they will actually be able to honour in practice. With a view to illustrating the scale of the problem facing performance engineers, we will briefly discuss some of the performance, resourcing and economic issues which affect an industry such as this. This will allow us to demonstrate directly the need for scalable performance analysis techniques.

Rarely are the needs of all customers of these services the same: some may require high processing performance but limited offline storage; others may require the ability to host massive data sets in memory and the CPU power to process these against incoming data in real time. It therefore makes economic sense to classify customers based on their stated service requirements and to charge them appropriately. Of course, these customers should expect some form of service guarantee, and the level of that guarantee is also likely to depend upon the price they are willing to pay. Quality of service guarantees are often phrased as *service level agreements (SLAs)* defined in terms of a probabilistic response-time upper bound. A single cloud com-

puting environment would ideally need to simultaneously satisfy tens or hundreds of different SLAs and would like to do so as inexpensively as possible.

For any cloud computing service provider to be economically viable, it is desirable to serve as many paying customers simultaneously as the infrastructure allows. Decisions must be made as to how the available physical resources can be optimally provisioned between different customers. There is not necessarily a large benefit to be had in massively exceeding one type of customer's SLA by allocating them unnecessarily powerful hardware when this could instead be used to take on additional customers. At the same time, it is crucial that the promises made to customers in the form of SLAs are maintained — the penalties for not doing so can be stiff both financially and in terms of customer satisfaction.

The vast number of different ways in which hardware can be allocated forms a massive, multi-dimensional parameter space; and the subset of this parameter space which simultaneously satisfies every customer's SLA determines a *feasible region*. By *feasible region*, we refer to the set of system configurations which work from the point of view of all customers. Even finding such an allocation can be difficult, or, in some cases, even impossible, but we would ideally like to go a step further and determine which feasible allocation is the *most efficient* or *best value* from the perspective of the service provider.

Exhaustively evaluating every possible allocation of resources to customers on a live cloud computing system is clearly impractical. Furthermore it does not facilitate testing the outcome of adding new physical resources to the infrastructure. If, however, an accurate mathematical performance model of the system can be designed, all of the possible resource allocations can be explored, and the optimum found, without actually requiring a physical or live version of the system. At least this is true in theory.

When using traditional performance evaluation techniques, the evaluation of even a *single* model configuration for a customer response-time profile can be computationally intractable even for a relatively modest system consisting only of hundreds of interacting entities. When coupled with the exceedingly multi-dimensional nature of an optimisation problem of the scale discussed above, it is clear that we require mathematical analysis techniques *orders of magnitude* faster than those currently available to performance engineers. The subject of this thesis is to develop a particular class of approximate analysis based on differential equations which can provide the speed-up that is necessary to realise the high-level goals alluded to above.

Traditional Markov chain-based modelling and analysis

The basic theoretical weapon in the performance engineer's arsenal is the *Markov chain*. The Markov chain is mathematically the simplest class of useful stochastic process but is also an extremely flexible modelling tool. In particular, more general discrete-state mathematical models not expressible as Markov chains can often be approximated very closely by one [e.g. 98, 196, 197]. The basic solution techniques for Markov chains are built on the application of algorithmically well-understood classical techniques taken directly from any linear algebra textbook [e.g. 177]. However, in spite of this seeming panacea, performance engineers have a big problem. It is often referred to as *state-space explosion* which alludes to an exponential growth in the *size* of the Markov chain with respect to the number of interacting components

in the system being modelled.[1]

In many respects this has not just become a problem recently — research into analysing large Markov chains has been active for many decades now. What is perhaps surprising though is that with the massive computational power available now to virtually anyone, let alone to performance engineers presumably with some form of budget, we are still struggling to solve useful models in a reasonable period of time. The reason for this is that the value of *large* that we are interested in is increasing, and it is increasing very fast. Such a growth is due in no small part to the desire to model recently emerged massively-parallel architectures such as peer-to-peer networks including file-sharing services and content-delivery networks; and cloud- or grid-computing scenarios like those discussed earlier where many thousands or even millions of participants interact; sometimes intentionally to complete some joint task, other times affecting others indirectly through contention for shared resources.

This massive state-space growth has rendered exact classical solution techniques all but useless for the extraction of quantitative performance measures from largely-proportioned models. This is true even when distributed solution techniques are employed [94]. Many attempts have been made to attack this problem such as the exact exploitation of model symmetry through state-space aggregation [e.g. 58, 112, 161, 222], however, at the scale of model we are interested in, this effectively just replaces one form of exponential growth with another.[2] Exact analytic, so-called, product-form solutions can sometimes be efficiently computed for certain restricted classes of queueing model [e.g. 34]. More recently, this approach has been extended to more general classes of models specified in, for example, *stochastic process algebra* [125, 127] and *stochastic automata networks* [126]. However even when the required conditions on the model parameters and structure do hold, this kind of analysis is not applicable to transient or time-bounded quantities.

Much progress has been made in the toleration of massive state spaces through using insightful data structures or compact symbolic representation schemes to encode the state space and dynamics of the model efficiently. The most widely used of these approaches are based on variants of binary decision diagrams [e.g. 103, 145, 165, 175] and have seen much prominence in the model checking community, including implementation in the popular tool *PRISM* [173, 174]. Whilst this representation works very well for storing the generator matrix of the Markov chain, it is less effective for storing the vector of state probabilities computed as the result of an analysis. In fact, a hybrid approach where the solution vector is stored explicitly in an array and only the generator matrix is represented symbolically appears now to be the most popular variant on this approach [172, 199]. This means that there is still surely at least a linear space and time dependence on the state-space size, meaning that, although very powerful, these techniques alone cannot completely address the issue of state-space explosion.

Discrete-event simulation is effectively the only feasible exact analysis approach for dealing with models of very large scale. Variants of the *Gillespie algorithm* [111] make stochastic simulation a viable approach in some situations. However, even simulation of Markov chains can quickly become expensive as the component population is increased — the cost of simulating a single trajectory grows at least linearly with respect to the number of individuals in the system. Furthermore, for a reasonable confidence interval many hundreds of replications are

[1] To get an idea of this problem consider a system consisting of N interacting individuals, each with D local states. If tracking each individual explicitly, the global state space could then be as large as D^N states.

[2] Indeed, for the example model mentioned above, the aggregated state space would have approximately at least $\exp(D)$ states if N is made large enough.

often required. This cost becomes even more pronounced if we wish to run many experiments over many different parameter combinations in order to solve an optimisation problem such as that mentioned earlier.

This thesis focuses on a class of techniques for efficiently approximating the stochastic dynamics of models with many interacting entities such as those discussed above. At its roots is the idea of approximating the evolution of the global component population by a suitable system of ordinary differential equations (ODEs). An equation is defined for each component state present in the model with the purpose of counting the number of components currently in that state. This technique results in massive analysis cost savings when applied to models consisting of many instances of a comparatively small number of component types. This is because the number of differential equations required is dependent only on the number of different component types and not on the actual number of individuals present in the model.

These ideas can be said to have had their origins in statistical physics where so-called mean-field models were constructed to approximate Markov chains representing the interactions between particles in plasma and dense gases [e.g. 26, 27, 117, 186]. Similar ideas were developed for chemical reaction networks [168] and also for epidemics [e.g. 7, 190]. More recently, this approach has been adapted to ameliorating the state-space explosion problem for models of computer and communication networks [e.g. 115, 116, 118] and protocols such as TCP [e.g. 12, 13].

Up until quite recently, most of these mean-field models applied to computer performance analysis have been constructed and reasoned about on a fairly ad-hoc, case-by-case basis. We are only now starting to see attempts to provide more unified frameworks and toolkits which might be applied with confidence by performance engineers without a substantial background in applied probability. Such approaches might include the discrete-time mean-field framework of Le Boudec et al. [179] and the derivation of ODE-based approximations from stochastic process algebra models as introduced by Hillston [147] for *Performance Evaluation Process Algebra (PEPA)* [146] and Bortolussi and Policriti [38] for *stochastic Concurrent Constraint Programming (sCCP)* [35].

1.2 Thesis summary

GPEPA and higher-order moment analysis In **Chapter 3**, we develop an extension of PEPA, *grouped PEPA (GPEPA)* [139], which is specifically designed to facilitate and generalise the existing ODE-based analysis approaches. As well as removing limitations regarding the class of models to which this kind of analysis can be applied, we show algebraically how the existing first-order approach can be viewed as an approximation to the first moments of component counts. This leads us to be able to define a qualitative predictor of the accuracy of the approximation. The main contribution of Chapter 3 is that we are able to generalise these ideas to construct analogous systems of differential equations which allow approximation of **higher-order moments** of component counts. Crucially, this means that we now have efficient access to, for example, **variance** information in massive performance models. This generalisation is not restricted just to PEPA, but could also be expressed straightforwardly in other frameworks such as the mean-field approaches of Le Boudec et al. [179], of Bobbio et al. [33] or of Bakhshi et al. [18]; and in other related areas such as *continuous stochastic Petri nets (CSPNs)* [86, 224].

Efficient extraction of passage-time measures The ability to access efficiently the moments of component counts is often something which can be applied directly to reason about the performance of a system. For example, we can vary a parameter and determine whether or not this increases or reduces the number of clients blocked waiting for a particular resource. However, as we have discussed, performance requirements such as *service level agreements (SLAs)* are most often specified in terms of *passage-time or response-time* statements, such as *a broadband connection must be established successfully within two seconds, 98% of the time,* rather than directly in terms of the number of components in some state at a given time. Extracting such passage-time measures using the efficient ODE-based approximation techniques introduced above is the subject of **Chapter 4**.

In particular, we consider two types of passage time. Firstly, we introduce the class of **global passage times**, which capture system-wide passages, that is, the time taken for a large proportion of components to complete some task or move into some state. An example of such a passage time might be the time taken for all peers to receive a file in a file sharing model. For this class of passage-time query, where the component population is large enough, we can provide a deterministic approximation. When the populations are smaller or there is more variance in the passage-time random variable, we can create bounds on the respective cumulative distribution function that can be computed efficiently using the approximate moment computations. Secondly, we consider **individual passage times** which capture the time taken for a single individual component to complete a task in the context of a larger model. For example, the time taken for a single node to exhaust its battery in a wireless sensor network model. For these passage-times, we are able to compute direct approximations to the respective cumulative distribution functions in both transient and steady-state cases.

Unified stochastic probes In addition to their quantitative evaluation, the accurate **specification** of detailed performance queries applied to large-scale performance models is itself a significant challenge. In **Chapter 5** we develop the **unified stochastic probe** framework for the combined specification and efficient ODE-based analysis of detailed performance queries in large-scale models. Unified stochastic probes are based on a regular expression syntax which is translated to a process algebra fragment which is then composed with the model to be interrogated. The resulting combined model is then analysed in order to evaluate the measure of interest. Importantly, this approach allows for the seamless capture of *derived behaviour*, i.e. that not necessarily present in the original state space of the model.

Convergence and error bounds for the first-order approximation Finally, in **Chapter 6**, we develop the theoretical underpinnings of the first-order ODE analysis exploited throughout this thesis. We utilise the powerful martingale methodology of Darling and Norris [85] to give **limiting convergence and error bounds** on the approximation in the transient case. Critically for many applications, we are also able to use techniques from stochastic approximation algorithms [29] to extend the **convergence results to the stationary distribution**. The stationary convergence result is dependent on first verifying certain asymptotic stability results for the system of ODEs — we also provide a powerful machinery which achieves this for a large class of GPEPA model. This approach can also be applied directly to other kinds of models with similar piecewise affine dynamics, including continuous stochastic Petri nets, certain kinds of queueing models where the number of servers scales jointly with the number of customers and also various other models of massively distributed computer systems [e.g. 205].

1.3 Applications and tool support

This thesis is a snapshot of a body of work which is already being put into practice. Indeed, as we have mentioned, in addition to simply being able to answer performance queries about models which would otherwise be inaccessible, one of the key benefits of this efficient ODE-based approach is the ability to jointly analyse many models, or parameterisations of models in a relatively short period of time. This opens the doors to performing the kind of optimisation analyses on massive models discussed at the beginning of this chapter. Preliminary work is already underway (although some of it beyond the explicit scope of this thesis) to exploit this. In Stefanek et al. [232] and Hayden et al. [142], we show how efficient ODE-based techniques can be exploited to perform fast optimisation over large multi-dimensional parameter spaces. Indeed, we also illustate this in Section 7.2 in the concluding chapter where we show how the rapid techniques developed in this thesis may be applied for efficient simultaneous verification of multiple service level agreements in a model of a massive customer–service system.

Preliminary support for parameter sweeping utilising ODE techniques is already available in the **GPA (grouped PEPA analyser)** tool[3] originally due to Stefanek et al. [229, 235], which implements many of the techniques described in this thesis. Furthermore, we have made available many of the models defined in this thesis on a website [131], both as pure differential equations in `MATLAB`® form and as GPEPA models in the syntax of the GPA tool.

1.4 Publications

I am co-author and made significant contribution to the following publications. Publications where I am first author are marked with *. The publications arose from the work conducted during the course of the research described in this thesis. In each case, we indicate how the paper fits into the narrative of the thesis, however not all of the work in these publications is described in detail in the thesis.

Journal articles

Fluid computation of the performance–energy trade-off in large scale Markov models *Accepted for publication in ACM SIGMETRICS Performance Evaluation Review, 2011.* Stefanek et al. [234]

This paper shows how ODE-based techniques can be used to investigate the trade-off between service level agreement satisfaction and complex reward optimisation. It is shown how ODE analysis naturally leads to a constrained global optimisation problem with embedded differential equations. This is also illustrated on an abstract model of a virtualised execution environment that accurately captures resource allocations.

[3] GPA is open source and the latest version can be downloaded from `http://code.google.com/p/gpanalyser`.

Mean-field approximations for performance models with generally-timed transitions.* *Accepted for publication in ACM SIGMETRICS Performance Evaluation Review, 2011.* Hayden [132]

This initial paper shows how the ODE-based approach for analysing Markovian models can be extended to incorporate generally-timed transitions. Specifically, in the context of a simple peer-to-peer software update model, it is shown how a system of delay differential equations can be formally derived from such models. It is verified empirically that the approximation is accurate and converges under the usual scaling. This work is outside the scope of this thesis and is only very briefly mentioned in the future work section of Chapter 7.

Fluid computation of passage time distributions in large Markov models.* *Accepted for publication in Theoretical Computer Science, 2011.* Hayden et al. [142].

This paper details how ODE-based approximation techniques can be used to extract key passage-time measures from massive performance models. Both passage-times which consider the global evolution of large populations of components; and those which consider the evolution of single tagged individuals are considered. Furthermore, questions of limiting convergence are addressed both in the transient and steady-state regime. Material from this paper constitutes most of the work in Chapter 4 and some of the work in Chapter 6.

A fluid analysis framework for a Markovian process algebra.* *Theoretical Computer Science, 411(22–24):2260–2297, May 2010.* Hayden and Bradley [139]

This paper presents the grouped PEPA (GPEPA) formalism and the general ODE-based moment approximation methodology significantly extending the original work by Hillston [147]. Material from this paper constitutes most of the work in Chapter 3.

Evaluating fluid semantics for passive stochastic process algebra cooperation.* *Performance Evaluation, 67(4):260–284, April 2010.* Hayden and Bradley [140]

This paper considers the extension of the ODE-based analysis approach to GPEPA models which utilise *passive cooperation*. Specifically, we show how a model involving passive cooperation can be translated to an equivalent model utilising only active cooperation. The ODE-based analysis approach can then be applied directly with much success. This work is not considered in detail in this thesis but is mentioned briefly in Chapter 3.

Submitted

Performance specification and evaluation with unified stochastic probes and fluid analysis.* *Submitted to Transactions on Software Engineering, September 2010.* Hayden et al. [141].

This paper develops the unified stochastic probe framework and shows how it can be used to specify key passage-time measures in massively-parallel computer systems. The analysis of these passage-time measures, using ODE-based methods, is then developed. Material from this paper constitutes most of the work in Chapter 5.

Conference papers

GPA - A tool for fluid scalability analysis of massively parallel systems. *In 8th International Conference on Quantitative Evaluation of Systems (QEST), Aachen, 2011.* Stefanek et al. [235]

This tool paper documents new functionality in the *Grouped PEPA Analyser (GPA)* tool. In particular, we discuss recently-developed extensions allowing specifications of complex reward measures using combinations of state based, rate accumulated and impulse rewards. Combined with the ability to efficiently capture various passage-time metrics, this paper shows how GPA can be used to solve optimisation problems with a reward objective function under different service level agreement type constraints. The details of how GPA works are outside the immediate scope of this thesis, although the numerical results appearing in the thesis were generated using the tool and, throughout the thesis, we will mention where GPA implements an analysis technique.

Fluid analysis of energy consumption using rewards in massively parallel Markov models. *In 2nd ACM/SPEC International Conference on Performance Engineering (ICPE), Karlsruhe, 2011.* Stefanek et al. [232]

This paper shows how efficient ODE-based analysis techniques can be used for the scalable analysis of energy consumption models. In order to achieve this, we extend the ODE approach to capture moments of accumulated rewards in Markov models. The trade-off between energy consumption and service level agreement compliance is investigated within this context. This work is outside the scope of this thesis and does not appear in detail in this thesis.

A new tool for the performance analysis of massively parallel computer systems. *In eighth workshop on Quantitative Aspects of Programming Languages (QAPL), pages 159–181, 2010.* Stefanek et al. [229]

This paper presents the *grouped PEPA analyser (GPA)* tool which implements ODE-based general moment approximations for the GPEPA formalism. It is also shown how the tool can be used to investigate the accuracy of the approximation. Material from this paper appears in Chapter 3.

Shared transaction Markov chains for fluid analysis of massively parallel systems.* *In 17th Annual Meeting of the IEEE International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), pages 1–12, 2009.* Hayden and Bradley [137]

This paper presents the *shared transaction Markov chain (STMC)* performance modelling formalism. The notion of a *shared transaction* between concurrent Markov chains is introduced. This allows a multi-phase synchronisation to accurately represent complex cooperation between modelling components in a compositional manner. We show how ODE-based analysis approaches may be extended to this context and discuss cases where this is not immediately possible. This work is outside the scope of this thesis and does not appear in detail, although

some of the ideas from this paper are considered in the future work section of Chapter 7 (Section 7.3.4).

Fluid semantics for passive stochastic process algebra cooperation.* *In 3rd International ICST Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS), 2008.* Hayden and Bradley [135]

This paper is a conference version of Hayden and Bradley [140].

Extracting Response Times from Fluid Analysis of Performance Models. *In SPEC International Performance Evaluation Workshop, pages 29–43, 2008.* Bradley et al. [52]

This paper presented preliminary work on the extraction of passage-time measures from massive performance models using ODE-based analysis techniques. Hayden et al. [142] and Chapter 4 of this thesis represent the current status of this work.

Submitted

Capturing the energy-performance trade-off in large-scale cloud computing environments. *Submitted to 3rd ACM/SPEC International Conference on Performance Engineering (ICPE), 2011.* Stefanek et al. [233]

This paper develops techniques for the accurate specification and efficient analysis of large-scale systems such as cloud-computing infrastructures with features such as multi-processor nodes and layered services. We extend the grouped PEPA formalism with channels to accurately capture multi-stage synchronisation between model components. Finally, we show how ODE-based approaches can be applied in this context to minimise energy consumption requirements constrained by multiple service level agreements for different customer classes. This work is outside the direct scope of this thesis and does not appear in detail, other than in the future work section of Chapter 7.

National workshop papers

The following papers appeared in various workshops at a national level and were not generally subject to rigorous peer review.

1. Hayden and Bradley [136] is concerned with the ODE-based general moment approximations for GPEPA described in Chapter 3 of this thesis;
2. Bradley et al. [53] is concerned with ODE-based passage-time approximation as described in Chapter 4;
3. Hayden and Bradley [138] is concerned with preliminary work to extract *stochastic differential equations* from PEPA models. This topic is not considered in detail in this thesis but is briefly mentioned in the future work section of Chapter 7 (Section 7.3.2);

4. Stefanek et al. [231] presents the grouped PEPA analyser (GPA) tool and its syntax and features;

5. Stefanek et al. [230] introduces an approach for the hybrid simulation/ODE analysis of PEPA models. This allows for simulation to be applied where the ODE approximation is likely to be less accurate;

6. Hayden [130] presented the initial steady-state convergence theorem for the ODE approximation. A refined version of this material appears in Chapter 6 of this thesis.

Statement of originality

I declare that this thesis was composed by myself, and that the work it presents is my own, except where otherwise stated.

Chapter 2

Background theory

2.1 Stochastic processes

In this thesis, we will be concerned with *Markov processes*, specifically, *continuous-time Markov chains (CTMCs)*, where *chain* here refers to the discrete nature of the underlying state space. In this section, we introduce the background theory related to CTMCs necessary to understand the rest of the thesis. Many standard definitions relating to stochastic processes and probability theory will be omitted for brevity, but the reader can find them in any standard text, for example, Kallenberg [157], Lawler [177] or Rogers and Williams [210].

As mentioned above, in this thesis, we will be interested in discrete-state stochastic processes. Let E be the finite set of states of the process. A *stochastic process* $X(t)$ for $t \in T$ is a family of E-valued random variables indexed by some set T. We will be interested primarily in the case $T = \mathbb{R}_+$ and we thus refer to the process as *continuous time*. A *Markov process* is a special type of stochastic process which models movement over time around E in a *memoryless* way. A formal definition follows.

Definition 2.1.1 (Markov process)**.** *A stochastic process $X(t)$ for $t \in \mathbb{R}_+$ with state space E is a Markov process if:*

$$\mathbb{E}[f(X(s+t))|\mathcal{F}_s] = \mathbb{E}[f(X(s+t))|X_s] \text{ a.s.}$$

for any $0 \leq s \leq t$ and f bounded, where $\mathcal{F}_t$ is the natural filtration of $X(t)$. This means that $\mathcal{F}_t$ encodes all information about the process $X(t)$ up until time t, for more details see, for example, Kallenberg [157].

A continuous-time Markov chain can be constructed in an elementary fashion from the so-called *jump chain* and sequence of *holding times* [85]. The jump chain is a sequence of E-valued random variables, say $Y(n)$ for $n \in \mathbb{Z}_+$, constructed such that:

$$\mathbb{P}(Y(0) = x_0, Y(1) = x_1, \ldots, Y(n) = x_n) = \pi(x_0)\pi(x_0, x_1)\ldots\pi(x_{n-1}, x_n)$$

in terms of some given *jump matrix* $\mathbf{\Pi} = (\pi(x, y) : x, y \in E)$ of probabilities, where $\pi(x, x) = 0$ for all $x \in E$; and *initial distribution vector* $\pi = (\pi(x) : x \in E)$. In other words $Y(n)$ is a

discrete-time Markov chain with transition matrix $\mathbf{\Pi}$. The holding times constitute a sequence S_n of non-negative real random variables, defined by:

$$S_n := T_n / q(Y(n-1))$$

where T_n is a sequence of independent and identically-distributed exponential random variables of parameter 1, independent also of the $Y(n)$, and $\mathbf{q} = (q(x) : x \in E)$ is the vector of non-negative *jump rates*. Note that S_n is thus itself an exponential random variable of rate parameter $q(Y(n-1))$. Finally we define the *jump time* random variables J_n by $J_0 := 0$ and $J_n := S_1 + \ldots + S_n$. We can then construct a continuous-time Markov chain $X(t)$ by setting $X(t) := Y(n)$ for $J_n \leq t < J_{n+1}$.

The matrix given by $\mathbf{Q} = (q(x,y) : x, y \in E)^T$, where:

$$q(x,y) := \begin{cases} q(x)\pi(x,y) & : x \neq y \\ -q(x) & : x = y \end{cases}$$

is called the *infinitesimal generator matrix* or $\mathbf{Q}$-*matrix*. Most of the time we will specify a particular CTMC of interest to us by giving its $\mathbf{Q}$-matrix and an initial distribution vector, from which it is clear a unique CTMC can be defined according to the construction given above. If we interpret the infinitesimal generator matrix $\mathbf{Q}$ as a linear operator acting on functions $f : E \to \mathbb{R}$ by $\mathbf{Q}f(x) := \sum_{y \in E} q(x,y) f(y)$, then it is possible to derive the following direct relationship between the evolution of the CTMC and $\mathbf{Q}$:

$$\begin{aligned} \frac{d}{dt}\mathbb{E}^z[f(X(t))] &= \lim_{u \to 0} \frac{\mathbb{E}^z[f(X(t+u))] - \mathbb{E}^z[f(X(t))]}{u} \\ &= \lim_{u \to 0} \frac{\mathbb{E}^z[\mathbb{E}[f(X(t+u)) - f(X(t)) | X(t)]]}{u} \\ &= \mathbb{E}^z[\mathbf{Q}f(X(t))] \end{aligned} \tag{2.1}$$

where $\mathbb{E}^z[\cdot]$ is the expectation taken with respect to the distribution of the Markov chain started in z, that is $\pi(z) = 1$. Taking $f(x) \equiv \mathbf{1}_{\{x=a\}}$ as the indicator function for some $a \in E$, this becomes:

$$\frac{d}{dt}\mathbb{P}^z\{X(t) = a\} = \mathbb{E}^z[q(X(t), a)] = \sum_{y \in E} \mathbb{P}^z\{X(t) = a\} q(y, a)$$

Then defining the vector of state probabilities $\mathbf{p}(t) := (\mathbb{P}\{X(t) = x\} : x \in E)$, we may write:

$$\dot{\mathbf{p}}(t) = \mathbf{Q}\mathbf{p}(t) \tag{2.2}$$

Equation 2.2 is often called the *Chapman–Kolmogorov forward equation* and describes the evolution of the state probabilities of the Markov chain. Solving this differential equation is the natural, most basic approach to analysing a Markov chain. We give in Figure 2.1 a very small example transition system which can be interpreted as defining a CTMC. Ordering the states numerically, the $\mathbf{Q}$-matrix for this CTMC is:[1]

$$\begin{pmatrix} -r_{t_a} & r_r & r_{t_b} & 0 & 0 & 0 \\ 0 & -r_r & 0 & r_{t_b} & 0 & 0 \\ 0 & 0 & -(r_{t_b} + r_{t_a}) & r_r & 2r_{t_b} & 0 \\ r_{t_a} & 0 & 0 & -(r_{t_b} + r_r) & 0 & 2r_{t_b} \\ 0 & 0 & 0 & 0 & -2r_{t_b} & r_r \\ 0 & 0 & r_{t_a} & 0 & 0 & -(2r_{t_b} + r_r) \end{pmatrix}$$

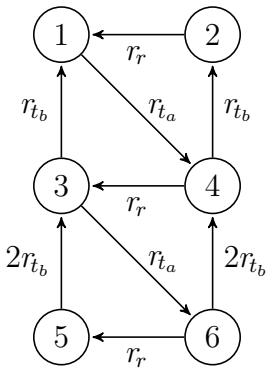


Figure 2.1: Simple labelled transition system which can be interpreted as a CTMC with state space $\{1, 2, 3, 4, 5, 6\}$ and transition rate between states equal to the value labelling the corresponding arc.

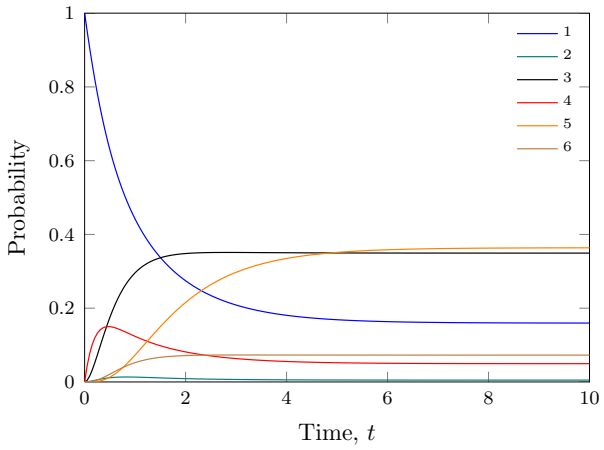


Figure 2.2: Solutions of the Chapman–Kolmogorov forward equations for each state corresponding to the CTMC given in Figure 2.1. Initial state is 1 and rates are: $r_{t_a} = 1.0$, $r_{t_b} = 0.4$, $r_r = 4.0$.

Since this CTMC is so small, it is computationally straightforward to solve its corresponding system of Chapman–Kolmogorov forward equations for the exact evolution of its state probabilities. Figure 2.2 shows the result of doing this. However, as we will see shortly, the dimensionality of most chains of interest makes this theoretically-straightforward approach extremely computationally prohibitive.

2.2 Solving continuous-time Markov chains

In an ideal world, we would be able to solve performance models quickly in an exact and fully algebraic manner. Even considering just the class of CTMC-based models of interest to us in this thesis, such a solution is available only for a very small class of models, for example, by means of symbolic Laplace transforms [251] for Markov chains with very few states or with an

[1]Note that the Chapman–Kolmogorov forward equations are sometimes written equivalently as $\dot{\mathbf{p}}^T(t) = \mathbf{p}^T(t)\mathbf{Q}^T$ so that then the row sums will be zero as opposed to the column sums.

extremely regular structure. Other than using product-form approaches (discussed briefly in Section 2.2.5 below), we are forced to turn to numerical techniques if we should expect any chance of being able to cope with systems of realistic scale. Unfortunately, we will see that even then, the traditional approaches to solving such performance models still do not scale tractably with realistically-proportioned systems, or can be applied only to a very restrictive class of models.

2.2.1 Naïve numerical solution of the underlying linear system

As described in Section 2.1, the evolution of the state probabilities of a continuous-time Markov chain is governed by a set of first-order linear ordinary differential equations, the *Chapman–Kolmogorov forward equations*. Specifically, $\dot{\mathbf{p}}(t) = \mathbf{Q}\mathbf{p}(t)$, where $\mathbf{p}(t)$ is the vector of marginal state probabilities at time t and $\mathbf{Q}$ is the infinitesimal generator matrix of the CTMC. The general solution to this equation is $\mathbf{p}(t) = \mathbf{p}(0)\exp(\mathbf{Q}t)$ where $\exp(\mathbf{Q}t)$ is the *matrix exponential* and is given by the following Taylor series [189]:

$$\exp(\mathbf{Q}t) = \sum_{i=0}^{\infty} \frac{(\mathbf{Q}t)^i}{i!}$$

It is well known that direct evaluation of the matrix exponential is numerically unstable since both positive and negative entries occur in the matrix $\mathbf{Q}$ [189]. An alternative method of solving for the transient state probabilities is to solve the system of ODEs numerically using standard techniques such as *Runge-Kutta*. However the scalability of this approach is very quickly hampered by the fact that the size of this ODE system is equal to the number of states which grows exponentially as components are added to the model (note that this statement also holds for direct evaluation of the matrix exponential).

The computational burden is lessoned somewhat if it is only steady-state probabilities which are of interest. In this case, we are interested in the fixed point $\mathbf{p}^*$ of the forward equation, obtained by setting $\mathbf{Q}\mathbf{p}^* = \mathbf{0}$. The solution of this homogeneous linear system is also a standard problem, considered for many tens of years in linear algebra. There exist standard approaches to solving such problems. Among the most efficient direct approaches is the construction of the LU factorisation of $\mathbf{Q}$ [113, 236], whose computation by Gaussian elimination can be made numerically stable [120]. The computational cost in terms of arithmetic operations of such an approach is dominated by that of the Gaussian elimination and is roughly cubic in the dimension of $\mathbf{Q}$, say n. However, recall that n itself grows exponentially as components are added to a model so for *increasing model size*, this approach experiences essentially exponential growth in complexity.

The problem $\mathbf{Q}\mathbf{p}^* = \mathbf{0}$ can also be tackled in an iterative fashion by one of the classical techniques, such as *Jacobi's method*, *Gauss–Seidel* or *successive over-relaxation* [212]. For very large matrices, these techniques often perform better than a direct approach. However it is harder to state their complexity since it is difficult to know in advance how many iterations it will take before a solution is within the desired accuracy window. Assuming that at least one iteration will be necessary, we can say that any iterative algorithm will require at least n arithmetic operations. As above, this still places our problem in an essentially exponential complexity class.

Many **Q**-matrices derived from CTMCs have the property of being *sparse*, that is, being populated mainly by zeros. Specialisations of both direct and iterative algorithms exist for sparse matrices which are often faster and more efficient in terms of their memory usage [212]. However, it remains true that in general we cannot do better than n arithmetic operations — we must surely still perform at least one arithmetic operation for each component of the solution vector.

For a concrete idea of how impractical such approaches are we might consider a fairly restrained CTMC model involving, say, 50 clients and 25 servers, each of which can be in one of five local states. Then the size of the state space is potentially $5^{75} \approx 10^{50}$, which also represents a rather coarse lower bound on the number of arithmetic operations required to solve this model for its steady-state distribution. At the time of writing, the world's fastest computer is the *Tianhe-1A* with a peak computing rate of around 2.5 petaFLOPS [237]. This computation would thus take at least 10^{34} seconds, or about 10^{26} years.[2]

2.2.2 Uniformisation

Uniformisation [119, 160, 193, 207] is a technique also based on the evaluation of an infinite series which has better numerical properties than the computation of the matrix exponential discussed in the previous section. The technique relies on the construction of an underlying discrete-time Markov chain (DTMC), the *uniformised DTMC*, which samples the CTMC at probabilistically regular time intervals. Let $\pi(n)$ be the marginal probability vector of this DTMC. It can be computed iteratively as:

$$\begin{aligned} \pi(0) &= \mathbf{p}(0) \\ \pi(i) &= \mathbf{P}\pi(i-1) \quad \text{for } i = 1, \ldots \end{aligned}$$

where $\mathbf{P} := \mathbf{Q}/q + \mathbf{I}$ and $q \geq \max_i |q_{ii}|$ is the *uniformisation rate* at which the CTMC is sampled. Then, by conditioning on the number of samplings which have occurred by time t, we may compute $\mathbf{p}(t)$ by the following infinite summation:

$$\mathbf{p}(t) = \sum_{i=0}^{\infty} \pi(i) \exp(-qt) \frac{(qt)^i}{i!}$$

Uniformisation has two significant benefits. Firstly, the error due to truncation can be bounded and thus a required error tolerance can be guaranteed. Secondly, since only multiplications and additions are involved in the evaluation of the infinite sum, it is a much more numerically stable algorithm than evaluating the matrix exponential. However, it still does not scale as a solution technique for realistic models. As above, it is still necessary to make at least n arithmetic computations in order to evaluate the sum since we must make at least one computation for each component of the solution vector.

2.2.3 State-space aggregation and lumpability

The techniques described above will work for any CTMC. Substantial improvements are to be had however by exploiting structure which may be present in CTMCs derived from particular

[2] This is a lower bound since it assumes some appropriate perfect parallel implementation suited to the massively heterogeneous architecture of this machine.

classes of model.

State space aggregation takes advantage of the fact that most models of computer and communication systems will include replicated identical components (for example, customers or servers in a network). Formally, given a continuous-time Markov chain $X(t)$ with state space E, and a partition of E, say $E = \uplus_{i=1}^{K} E_i$, we may construct another *aggregated* stochastic process, say $Y(t)$, taking values in $\mathcal{E} := \{E_i \ : \ 1 \leq i \leq K\}$ defined by $Y(t) := E_i$ if $X(t) \in E_i$ for any i. If the new process $Y(t)$ is itself a Markov process for any initial distribution, it is called *(ordinarily) lumpable*. This can be characterised in terms of the transition rates [161]. Indeed, $X(t)$ is lumpable with respect to the partition $\mathcal{E}$ if and only if, for any E_i, $E_j \in \mathcal{E}$ and $x, y \in E_i$:

$$\sum_{z \in E_j} q(x, z) = \sum_{z \in E_j} q(y, z) \tag{2.3}$$

If this holds, Equation 2.3 is then the transition rate from state E_i to E_j for the lumped process $Y(t)$. This, hopefully smaller, Markov chain can then be analysed in place of the larger one. In return for the computational savings this may afford, one is able only to obtain probabilistic information about aggregated quantities. Another notion of lumpability, *exact lumpability*, also exists [216]. This is usually more difficult to satisfy but can provide information about the original unaggregated process.

Examples of the use of lumpability include direct application to the Markov chain [58] and the application to a hierarchically structured Markovian state-space [57, 222]. Additionally, lumpability can be mechanically derived from models which display suitable component replication and are specified in higher level formalisms such as stochastic Petri nets [71] and stochastic process algebras [112].

In this thesis, we are particularly interested in models which include many replicated identical components as constructed to represent computer systems exhibiting massive parallelism. In this case, state-space aggregation, in the form of ordinary lumpability, can be applied naturally to construct a CTMC where the states encode the number of components of a particular type that are in a particular local state. This is as opposed to representing each component's state individually. Such models are often termed *population models*. Queueing network models are traditionally given in this form — for example, the number of customers at a queueing node is recorded as opposed to explicitly recording this fact individually for each of the customers. In the case of the stochastic process algebra PEPA, an automatic algorithm for aggregating in this manner has been presented by Gilmore et al. [112], which derives a so-called *numerical vector form* for PEPA models.

While this kind of approach is effective in reducing the state space, it does not even come close to eliminating the problem of state-space explosion. To see this, let us consider the simple case of a system which involves N interacting components, each of which can be in one of D local states. When representing this as a CTMC, without any aggregation, we would have D^N states to contend with. Under the aggregation discussed above, where we maintain counts of the number of the N components in each of the D local states, we will have $\binom{D+N-1}{N}$ states. For large enough N, this is bounded below by $\exp(D)$ (usually very loosely) so models of any detail will quickly result in massive aggregated state spaces. For example, if $N = 150$ and $D = 15$, we already have an aggregated state space with almost 10^{20} states.

The kinds of aggregations discussed above are exact in the sense that the aggregated process does correctly capture probabilistic behaviour related to the original CTMC. When exact

aggregations are not possible, it is often the case that a CTMC partition can be considered *almost lumpable* in some sense. Formally, the definition usually adopted here is that of *quasi-lumpability* [101]. This can be used to perform *approximate abstraction* but it is very difficult in general to reason about the error that is introduced by doing this, although there has been some progress in bounding the stationary probabilities for nearly-decomposable chains [88]. *Abstract Markov chains* [159] compute an interval in which the transition rates between aggregated states must lie, defining a set of Markov chains. Model checking can then be performed using a three-valued version of continuous stochastic logic (CSL). On the other hand, the technique of *stochastic bounds* [238] attempts to change the rates of an existing CTMC so as to make it exactly lumpable. The difficulty is in ensuring that the modification guarantees an upper or lower bound on the performance quantity of interest. We also note here that Smith [228] has investigated the *compositional* application of such approximate aggregation techniques in the context of the process algebra PEPA.

Unfortunately, the class of models to which these kind of approximate aggregation techniques can be applied is fairly restrictive. Even when they can be applied, this author is certainly not aware of any work showing how such techniques are able to reduce the state space for massively-parallel models by the exponential orders of magnitude required to obtain tractability. Furthermore, it is hard to predict in general whether the bounds obtained by performing such techniques will actually be tight enough to provide useful performance information.

2.2.4 Model representation techniques for more efficient solution

The *tensor representation* has been used for a long time now as a means to reduce the memory required to encode the state space of a performance model. It was originally presented by Plateau [203] in the context of *stochastic automata networks (SANs)* [202]. Whilst in some cases, this can achieve substantial memory savings, the space efficiency is obtained at the expense of an increased computation time. For this reason, this approach cannot completely address the state-space explosion problem since the computation time to perform the Markov chain analysis, as well as the memory requirements, grow exponentially as components are added to the model.

Tools such as *PRISM* [171], *Möbius* [90] (in symbolic mode) and *SMART* [74] employ a *multi-terminal binary decision diagram (MTBDD)* [e.g. 103, 145, 165, 175] or *matrix diagram* [e.g. 76] symbolic representation to store the state space of the CTMC underlying a model. These approaches can potentially handle very large models in only small amounts of memory. Unfortunately, the performance of numerical calculations made using these representations can often be hampered by an explosion in the number of distinct values in the probability vector being computed [e.g. 89, 123, 158]. As already mentioned, this has led to a hybrid approach being adopted, where the solution vector is stored explicitly in an array and only the generator matrix is represented symbolically [172, 199]. This means that there is still at least a linear growth in the space and time complexity of such an approach with respect to the size of the state space.

2.2.5 Product-form solution and mean value analysis

In an ideal world, a model defined compositionally could also be solved compositionally. That is, a massively parallel model could be handled by analysing the individual constituents of the model in isolation and then combining the results in an inexpensive fashion to obtain global information regarding the composed system.

In this vein, it can be shown that certain classes of performance models enjoy a *product-form solution*, which means that the distribution of the entire system at steady-state can be expressed as a product of the marginal steady-state distributions of the constituent components of the system. The conditions for this to be true are quite restrictive. However, a number of the key results in queueing theory pertain to the existence of product-forms in, for example, Jackson [154] and BCMP [25] queueing networks. The most recent research in this area has been concerned with generalising the idea of product forms beyond just queueing networks into the realm of more general models such as those specified in stochastic process algebra [125]. Product forms have also been exploited to perform efficient model checking of statements posed in *continuous stochastic logic (CSL)* [22, 23]. Although very powerful where they do exist, product-form results are unfortunately relatively rare and do not offer information about the transient evolution of a performance model. Furthermore, even when they can be found, it can still be numerically expensive to work out exactly the form of the product — finding the correct normalising constant can often be a costly process.

A related technique which emerged from this area of queueing theory research is *mean value analysis (MVA)* [208], which is built upon the *arrival theorem* of Lavenberg and Reiser [176], Sevcik and Mitrani [219]. This gives a direct algorithm for deriving average steady-state performance metrics from a product-form queueing network. It is useful for networks where the computation of the normalising constant is too expensive. Although usually cheaper than a direct Markov chain analysis, the complexity of this approach does grow linearly with the population size and can thus become expensive for large component populations. There are also many further generalisations and approximations of the original MVA approach including to non-product-form networks [e.g. 34]. Very recently, Thomas and Zhao [245] have explored the generalisation of the MVA approach to the stochastic process algebra setting with some success.

2.3 PEPA

2.3.1 Introduction

Performance Evaluation Process Algebra (PEPA) [146], is a simple stochastic process algebra, but one which has sufficient expressiveness to model a wide variety of systems, including multimedia applications [46], mobile phone usage [100], GRID scheduling [246], production cell efficiency [148] and web server clusters [50, 54] amongst others. It is also adept at capturing large parallel software systems, such as peer-to-peer networks [96], to which the style of analysis considered in later chapters is particularly well suited.

As in all process algebras, systems are represented in PEPA as the composition of *components* which undertake *actions*. Associated with each timed action is a duration, specified by a rate parameter. So, for example, the expression $(\alpha, r).P$ denotes a component which can undertake

a timed α-action, at rate r to evolve into a component P. The rate r is interpreted as a random delay sampled from an exponential distribution with parameter r. This means that the stochastic behaviour of the model is governed by an underlying continuous-time Markov chain, the explicit definition of which will be given later in this section.

2.3.2 Syntax

PEPA has a small set of combinators allowing system descriptions to be built up as the concurrent execution and interaction of simple sequential components. The syntax of a PEPA component may be specified formally using the grammar:

$$\begin{aligned} S &::= (\alpha, r).S \;\mid\; S + S \;\mid\; \boldsymbol{C}_S \\ P &::= P \underset{L}{\bowtie} P \;\mid\; P/L \;\mid\; S \;\mid\; \boldsymbol{C}_P \end{aligned} \tag{2.4}$$

where α is drawn from a set $\mathcal{A}_t$ of *timed action types*, which always includes the distinguished *hidden action type* τ. A *cooperation set* L is a subset of $\mathcal{A}_t \backslash \{\tau\}$ and $r \in \mathbb{R}_+ \cup \{n\top \mid n \in \mathbb{Q}, n > 0\}$ is a rate parameter. A rate in $\mathbb{R}_+$ represents an *active* enabling of an action and a rate of the form $n\top$ represents a *passive* enabling of an action. A timed transition (α, r) is also referred to as a *timed activity*.

A PEPA component can be a purely *sequential component* S or a *model component* P with its own internal parallelism and we write $\mathcal{C}$ for the set of all PEPA components. $\boldsymbol{C}_S$ and $\boldsymbol{C}_P$ stand for a constant which denotes a sequential component or a model component respectively, as introduced by a definition. The effect of this syntactic separation between constants is to allow cooperation between sequential components only.

2.3.3 Semantics

A discussion of the basic operators and cooperation framework for PEPA is given below before we provide the formal operational semantics (Figure 2.3).

Prefix The basic mechanism for describing the behaviour of a system with a PEPA model 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 with rate r, and it subsequently behaves thereafter as P.

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 behaviour 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 behaviour 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 also allows the recursive definition of components, for example, $\boldsymbol{X} \stackrel{def}{=} (\alpha, r).\boldsymbol{X}$ performs α at rate r repeatedly.

Hiding The possibility to abstract away some aspects of the behaviour of a component is provided by the hiding operator, denoted P/L. Here, the set L identifies those action types which are to be considered internal or private to the component and which will appear as the hidden action type τ in the transition system of the model.

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 synchronise. For action types not in L, the components proceed independently and concurrently with their enabled activities. We write $P \parallel Q$ as an abbreviation for $P \underset{\emptyset}{\bowtie} Q$, where P and Q execute independently in parallel.

If a cooperating component enables an activity whose action type is in the cooperation set it will not be 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 activity has to be altered to reflect the slower component in a cooperation. Within the cooperation framework, PEPA assumes *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 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. All passive actions must be synchronised in a final (outermost) component and a component is not allowed to enable the same timed action type both passively and actively. In order to support passive enabling of actions in the calculation of rates, we require the following algebraic rules:

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

The *apparent rate* $r_\alpha(P)$ calculates the observed rate at which a process 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. A formal definition follows.

Definition 2.3.1 (PEPA :: Apparent rate)**.** *Let P be a PEPA component. Let $\alpha \in \mathcal{A}_t$ be a timed action type. Then the apparent rate is $r_\alpha(P)$, defined as follows.*

$$\begin{aligned} r_\alpha((\beta, r).P) &:= \begin{cases} r & \text{if } \beta = \alpha \\ 0 & \text{if } \beta \neq \alpha \end{cases} \\ r_\alpha(P + Q) &:= r_\alpha(P) + r_\alpha(Q) \\ r_\alpha(P/L) &:= \begin{cases} r_\alpha(P) & \text{if } \alpha \notin L \\ 0 & \text{if } \alpha \in L \end{cases} \\ r_\alpha(P \underset{L}{\bowtie} Q) &:= \begin{cases} \min(r_\alpha(P), r_\alpha(Q)) & \text{if } \alpha \in L \\ r_\alpha(P) + r_\alpha(Q) & \text{if } \alpha \notin L \end{cases} \end{aligned}$$

Figure 2.3 presents the structured operational semantics for PEPA. This determines a transition relation on PEPA components. For a PEPA component P we define its set of *derivative states* $ds(P)$ to be all components reachable from it by timed transitions according to the operational semantics. That is, $ds(P)$ is the smallest set of PEPA components such that $P \in ds(P)$ and if for any $P_1 \in ds(P)$, $P_1 \xrightarrow{(\cdot,\cdot)} P_2$[†] then $P_2 \in ds(P)$. Also if $\mathsf{P} \subseteq ds(P)$, we define $ds(\mathsf{P}) := \cup_{Q \in \mathsf{P}}\{ds(Q)\}$. In PEPA, multiply-enabled activites will induce multiple transitions in the underlying Markov chain, so we define the *multiset* of enabled activities of a PEPA component P to be $\mathcal{A}ct(P) := \{\!|(\alpha, r) : P \xrightarrow{(\alpha,r)} |\!\}$. Furthemore, we write $\mathcal{A}_t(P)$ for the set of timed action types which are used by any derivative state of P, that is, $\mathcal{A}_t(P) := \cup_{P' \in ds(P)}\{\alpha : P' \xrightarrow{(\alpha,\cdot)}\}$.

For a given PEPA component P we may naturally construct its *derivation graph*, a labelled and directed multigraph. The nodes of this multigraph are the derivative states of P, that is, the set of nodes is $ds(P)$. Two nodes in the multigraph, say P_1 and $P_2 \in ds(P)$, have a directed arc between them for every transition $P_1 \xrightarrow{(\alpha,r)} P_2$. The label of this arc is then the activity corresponding to the transition, that is, (α, r).

The derivation graph can then be interpreted naturally as a continuous-time Markov chain, whose states are given by the nodes (i.e. derivative states) and each arc represents a transition at the rate of the activity labelling the arc. We call this the *underlying continuous-time Markov chain* of the model.

2.3.4 Example

We will now give a simple example of a performance model in PEPA representing the abstract scenario of many concurrent processors performing some job, part of which requires the use of some resource taken from a limited pool. The action $task_a$ will represent some initial task which the processors must perform in conjunction with a resource. The action $task_b$ will represent a task then performed by processors independently of resources. Resources will also be required to perform an independent *reset* action after they have been used by a processor. This model can be represented in PEPA as $\boldsymbol{PR}(n, m)$ (n processors and m resources) defined by the equations in Figure 2.4, where we define the shorthand for parallel composition of k components by $C[k] := \underbrace{(C \parallel \ldots \parallel C)}_{k}$.

We note here that the CTMC given in Figure 2.1 is that obtained by aggregating the PEPA model $\boldsymbol{PR}(2, 1)$ as described in Section 2.2.3, where state $i \in \{1, 2, 3, 4, 5, 6\}$ in Figure 2.1 corresponds to the aggregate state of $\lfloor \frac{6-i}{2} \rfloor \times \boldsymbol{Proc}_0$ components, $(2 - \lfloor \frac{6-i}{2} \rfloor) \times \boldsymbol{Proc}_1$ components, $(i \mod 2) \times \boldsymbol{Res}_0$ components and $(i - 1 \mod 2) \times \boldsymbol{Res}_1$ components.

2.4 Related work

In this section we survey CTMC-analysis techniques which are closely related to the topic of this thesis.

[†]We write $P \xrightarrow{(\cdot,\cdot)} Q$ to stand for $P \xrightarrow{(\alpha,r)} Q$ where the existence of the transition is all that matters and the parameters α and r are not relevant.

Prefix

$$\frac{}{(\alpha, r).E \xrightarrow{(\alpha,r)} E}$$

Competitive Choice

$$\frac{E \xrightarrow{(\alpha,r)} E'}{E + F \xrightarrow{(\alpha,r)} E'} \qquad \frac{F \xrightarrow{(\alpha,r)} F'}{E + F \xrightarrow{(\alpha,r)} F'}$$

Cooperation

$$\frac{E \xrightarrow{(\alpha,r)} E'}{E \underset{S}{\bowtie} F \xrightarrow{(\alpha,r)} E' \underset{S}{\bowtie} F} \quad (\alpha \notin S) \qquad \frac{F \xrightarrow{(\alpha,r)} F'}{E \underset{S}{\bowtie} F \xrightarrow{(\alpha,r)} E \underset{S}{\bowtie} F'} \quad (\alpha \notin S)$$

$$\frac{E \xrightarrow{(\alpha,r_1)} E' \quad F \xrightarrow{(\alpha,r_2)} F'}{E \underset{S}{\bowtie} F \xrightarrow{(\alpha,R)} E' \underset{S}{\bowtie} F'} \quad (\alpha \in S)$$

where $R := \frac{r_1}{r_\alpha(E)} \frac{r_2}{r_\alpha(F)} \min(r_\alpha(E), r_\alpha(F))$

Hiding

$$\frac{E \xrightarrow{(\alpha,r)} E'}{E/H \xrightarrow{(\alpha,r)} E'/H} \quad (\alpha \notin H) \qquad \frac{E \xrightarrow{(\alpha,r)} E'}{E/H \xrightarrow{(\tau,r)} E'/H} \quad (\alpha \in H)$$

Constant

$$\frac{E \xrightarrow{(\alpha,r)} E'}{\boldsymbol{C} \xrightarrow{(\alpha,r)} E'} \quad (\boldsymbol{C} \stackrel{def}{=} E)$$

Figure 2.3: Structured operational semantics of PEPA

$$\boldsymbol{Proc}_0 \stackrel{def}{=} (task_a, r_{t_a}).\boldsymbol{Proc}_1 \qquad \boldsymbol{Res}_0 \stackrel{def}{=} (task_a, r_{t_a}).\boldsymbol{Res}_1$$
$$\boldsymbol{Proc}_1 \stackrel{def}{=} (task_b, r_{t_b}).\boldsymbol{Proc}_0 \qquad \boldsymbol{Res}_1 \stackrel{def}{=} (reset, r_r).\boldsymbol{Res}_0$$

$$\boldsymbol{PR}(n, m) \stackrel{def}{=} \boldsymbol{Proc}_0[n] \underset{\{task_a\}}{\bowtie} \boldsymbol{Res}_0[m]$$

Figure 2.4: PEPA description of the processor–resource model $\boldsymbol{PR}(n, m)$.

2.4.1 *Fluid-flow* analysis of PEPA models

In the context of PEPA, the first application of approximate ODE-based analysis techniques was made by Hillston [147] and referred to as *fluid-flow analysis*.[3] Specifically, it was shown how a system of differential equations could be derived heuristically from a suitable PEPA model. The idea was then that the continuous solutions of these ODEs could be taken, in some sense, to approximate the model's underlying discrete CTMC. It should be noted that this work built upon that of Sumpter et al. [239] for the discrete-time process algebra *WSCCS* [247].

This approach is best suited to models with a high degree of parallelism between sets of identical components which may cooperate with other such sets of components over a set of shared actions. One such example would be the PEPA model $\boldsymbol{PR}(n, m)$ given earlier in Section 2.3.4, which, for large n and m, models a large group of identical processors interacting with a large group of identical resources. For this example, the approach of Hillston [147] results in the construction of *continuous* real-valued quantities, say $\boldsymbol{P}_0(t)$, $\boldsymbol{P}_1(t)$, $\boldsymbol{R}_0(t)$ and $\boldsymbol{R}_1(t)$, each of which is intended as a *deterministic* approximation to the *stochastic process* which maintains the *discrete count* of the number of instances of the respective sequential component present within the model at time t. To derive the approximating system of differential equations, the change in a component count over a short interval of time δt is considered. This can be expressed approximately in an intuitive manner by considering the aggregate transition rate at which sequential components of this type enter the model minus that at which they exit. For the $\boldsymbol{Proc}_0$ component, such considerations lead to the following approximate equation:

$$\boldsymbol{P}_0(t+\delta t) - \boldsymbol{P}_0(t) \approx -\min(\boldsymbol{P}_0(t), \boldsymbol{R}_0(t))r_{t_a}\delta t + \boldsymbol{P}_1(t)r_{t_b}\delta t$$

Dividing by δt, taking the limit as $\delta t \to 0$ and proceeding similarly for the other three sequential components leads to the following system of differential equations:

$$\begin{aligned}
\dot{\boldsymbol{P}}_0(t) &= -\min(\boldsymbol{P}_0(t), \boldsymbol{R}_0(t))r_{t_a} + \boldsymbol{P}_1(t)r_{t_b} \\
\dot{\boldsymbol{P}}_1(t) &= -\boldsymbol{P}_1(t)r_{t_b} + \min(\boldsymbol{P}_0(t), \boldsymbol{R}_0(t))r_{t_a} \\
\dot{\boldsymbol{R}}_0(t) &= -\min(\boldsymbol{P}_0(t), \boldsymbol{R}_0(t))r_{t_a} + \boldsymbol{R}_1(t)r_r \\
\dot{\boldsymbol{R}}_1(t) &= -\boldsymbol{R}_1(t)r_r + \min(\boldsymbol{P}_0(t), \boldsymbol{R}_0(t))r_{t_a}
\end{aligned}$$

Figure 2.5 shows one solution of these ODEs for the particular set of initial conditions corresponding to the initial state of the model $\boldsymbol{PR}(100, 40)$. The power of this approach comes from the fact that the analysis of a model with a potentially unbounded state space (its size is an unbounded function of n and m) has been reduced to the solution of a fixed and usually small set of ODEs. This is paid for in that the analysis is only approximate and its relationship to the original CTMC is not immediately clear. Furthermore, the ODE solutions as they stand provide an approximate view of the component count at a given time. In the field of biochemical modelling, raw component counts are useful for capturing fluctuations in concentrations. This has been heavily exploited in biologically-oriented process algebras such as *BioPEPA* [77] and *stochastic π-calculus* [62]. For models of computer, communication and business processes, measures such as passage-time distributions are also very useful. The generation of passage-time metrics, such as means, quantiles and SLAs, from ODE-based analysis is discussed in Hayden et al. [142], Tribastone [249] and Chapter 4 of this thesis.

[3] The term *fluid-flow analysis* is used in the literature by various authors. We believe that *differential equation-based analysis* is clearer and use this term throughout the rest of this thesis unless referring explicitly to literature which uses the former term.

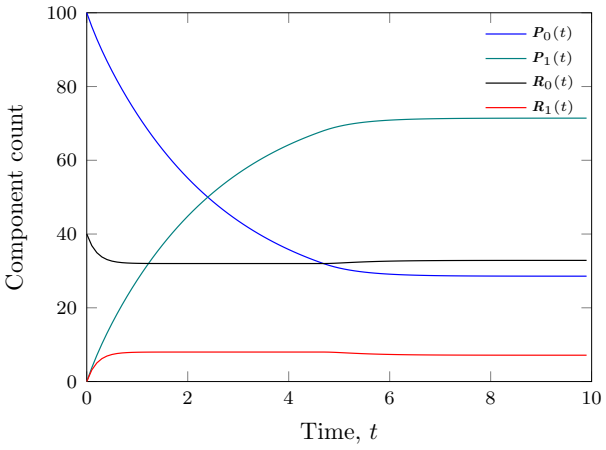


Figure 2.5: ODE solutions for the PEPA model $\boldsymbol{PR}(100, 40)$. Rates: $r_{t_a} = 1.0$, $r_{t_b} = 0.4$, $r_r = 4.0$.

In the early work of Hillston [147], there were also limitations imposed on the class of models which could be subject to such analysis:

1. Firstly, the technique was only applicable to models where the large groups of parallel components to be approximated by ODEs consisted only of sequential components. It was not shown how differential equations could be constructed to approximate the count of more general PEPA components (non-sequential) present in a model;

2. These sequential components were forced to cooperate over all common action types;

3. The same action type was not allowed to be enabled by two distinct local states of the same sequential component;

4. Synchronising sequential components were forced to enable shared actions at the same rate;

5. Action hiding was not considered.

The last four of these limitations have since been addressed in generalised versions of the differential equation-based approach given by various authors [51, 91, 139, 140, 249, 250]. This approach for the analysis of PEPA models has also been applied in many diverse application areas including, for example, peer-to-peer networks [96], non-repudiation protocols [257], Internet worms [51], web services [82], emergency egress [182], content-adaptation systems [91, 92], ubiquitous systems [124], key distribution centres [81, 258] and energy consumption [232].

Furthermore, progress has also been made in terms of relating the differential equation approximation to the PEPA model's underlying CTMC. In particular, in his thesis, Tribastone [249] (see also Tribastone et al. [250]) applies a result of Kurtz [166] to show that in the limit of large component populations, the suitably rescaled CTMCs converge to the solution of the approximating ODEs in probability over bounded intervals of time. Similar results have also been given by Ding [91] in his thesis, Geisweiller et al. [108] and Hayden and Bradley [140].

Ding [91] has also considered the analogous question of convergence in the stationary or steady-state regime. However, as we discuss in detail in Section 6.3.3, his general result represents a theoretical guarantee which can only be applied to models with a tractable state space. In order to apply it, one first has to verify a precondition for the specific sequence of PEPA models of interest which itself suffers from the state-space explosion problem since it depends directly on the steady-state probabilities of the underlying CTMCs. Guarantees just that the ODEs have a fixed point to which all trajectories eventually converge are given for a very simple class of model.

Tribastone [249] has also shown how average steady-state response times in PEPA models may be computed by combining the differential equation analysis with the classical result from queueing theory known as Little's Law [181]. Such an approach has also been applied by Ding [91] and is used by Clark et al. [81] and Kesidis et al. [162]. In addition to average response time, Tribastone [249] has also considered the differential equation-based analysis of the quantities *action throughput* (the total rate at which a particular action is performed) and *capacity utilisation* (measures the actual use of a component to perform a particular action as a proportion of its maximum capacity to perform it). Differential equation analysis of similar quantities, again termed *throughput* and *utilisation*, have also been considered by Ding [91].

Finally, we note that a hybrid approach to differential equation analysis for PEPA has also been proposed by Bortolussi et al. [45]. This is based on that already defined for the *stochastic Concurrent Constraint Programming (sCCP)* process algebra [35] in Bortolussi and Policriti [44], which is discussed in more detail in the next section.

2.4.2 Differential equation-based approaches for sCCP and π-calculus

Stochastic concurrent constraint programming (sCCP) [35] is a Markovian stochastic extension of *concurrent constraint programming (CCP)* [214], a modelling language which is essentially a stochastic process algebra augmented with a constraint store. Synchronisation occurs indirectly via this constraint store rather than directly between process algebra terms.

Motivated primarily by solving models in biochemistry, Bortolussi and Policriti [38] give the translation of an sCCP program into a system of differential equations (as well as an inverse procedure). The approach taken is analogous to that of Hillston [147] for PEPA. It is also shown that the translation given for sCCP *preserves rate semantics* which means here that a biochemical model in sCCP yields a system of ODEs which matches the traditional view taken by biochemists. In Bortolussi and Policriti [41], a more detailed presentation of the above is given together with many examples taken from biochemistry. In the context of these examples, it is suggested that asymptotic stability of a fixed point is necessary for convergence of the CTMCs to the ODE solution in the steady-state regime, although this is not considered or shown formally as we do in Section 6.3.3. In the bounded time interval case, the aforementioned result of Kurtz [166] is used to obtain a convergence result analogous to that for PEPA discussed in the previous section.

Cardelli [62, 63] has previously presented various translations from subsets of the stochastic process algebra *stochastic π-calculus* [204] to systems of chemical reactions, and back. Furthermore, these translations are shown to preserve discrete (CTMC-based) semantics and continuous (ODE-based) semantics. In Cardelli [61], it is shown that under a natural translation

from a subset of π-calculus to a system of chemical reactions, the dynamics of the underlying CTMCs are preserved by showing that the respective Chapman–Kolmogorov forward equations are equivalent. However, the relationship between the discrete semantics of the CTMC and continuous semantics of the differential equations is not considered explicitly.

In Bortolussi [36], the differential-equation approach for sCCP is extended through the definition of ODEs approximating second-order moments. These are derived from the Chapman–Kolmorogov forward equations which govern the model's underlying CTMC, referred to in that work as the *master equation* — a term often used when considering chemistry models. This approach is mathematically equivalent to that applied earlier by Hayden [129] in his undergraduate thesis (see also Hayden and Bradley [134, 136]) which allowed the derivation of ODEs approximating higher-order moments of component counts in PEPA models.

Bortolussi and Policriti [39, 40, 42, 43, 44] have also investigated hybrid approximation techniques which can be seen as an approach lying between direct analysis of the underlying CTMC of a model and its ODE approximation. These are useful where there may be component populations which are small enough so that their discrete nature is not well approximated by the continuous differential-equation approximation. This approach consists of translating an sCCP model to a *transition-driven stochastic hybrid automaton (TDSHA)* [59] which is essentially a finite automaton augmented with a set of variables which evolve continuously in each automaton state according to a set of state-dependent differential equations. The stochastic nature of such a model is due to the fact that the automaton state changes are governed stochastically in a Markovian fashion. The underlying stochastic process of such a model is a *piecewise-deterministic Markov process (PDMP)* [87].

The discrete dynamics is given by the transitions between the automaton states so that they capture the state of discrete model components rather than the differential equations. However, the differential equations still play the important role of capturing the state of components which are deemed numerous enough. In this manner, such a representation would still represent a saving in terms of state-space size over the completely discrete underlying CTMC. This being said, the only tractable general method for analysing PDMPs is by employing stochastic simulation which is likely to be much more costly than employing the purely ODE-based approach, even if it is cheaper than simulation of the entire underlying CTMC. When timing information is not important, there are, in some cases, alternatives to simulation. For example, Chen et al. [70] have shown how *untimed reachability probabilities* on PDMPs can be found by solving sets of *Volterra integral equations*, approximated by systems of partial differential equations.

Finally, very recently, Bortolussi [37] was able to prove a bounded-time convergence result of an sCCP program's underlying CTMC to the solution of the associated PDMP as the size of the continuously-approximated component populations are increased. We do not discuss hybrid approximation approaches such as this further in this thesis — instead our focus here is purely on the evolution and development of the ODE-based approach.

2.4.3 Mean-field models for computer and communication systems

The *mean-field approach* bears similarities to the approaches discussed above for stochastic process algebra. The mean-field idea was first introduced in physics whereby complicated interactions between particles are, in some sense, replaced by an averaged quantity, the *mean*

field, which allows for a tractable view of the system's dynamics. Such an approach can be applied, for example, to models of plasma and dense gases where the interaction between particles decays suitably as the system size is increased (see e.g. Baxter [26], Bellomo and Pulvirenti [27], Graham and Méléard [117], Méléard [186]). More recently this approach has found application in the analysis of queueing and communication systems [115, 116, 118, 185, 211, 220, 248] and the TCP protocol [12, 13], to give a few computing examples.

Most of the approaches cited in the preceding paragraph rely on fairly ad-hoc theory and proofs. More closely related to our work however is the mean-field framework introduced by Le Boudec et al. [179] (see also Benaïm and Le Boudec [30], Cho et al. [72, 73], Freudiger et al. [102], Gómez-Serrano et al. [114], Ioannidis [150], Ioannidis and Marbach [151], Ioannidis and Massoulié [152], Ioannidis et al. [153], Mundinger and Le Boudec [192], Zyba et al. [259]). Although the models in this framework are in *discrete time*, the particular model dynamics considered result in a limiting *continuous-time* differential-equation approximation. This approach is essentially mathematically analogous to the ODE-based analyses described above for PEPA and other formalisms except that in the cases above, we start with continuous-time models rather than discrete time. Indeed, the finite-time convergence results proved by Le Boudec et al. [179] have a very similar form and implication to those proved in the case of the ODE-based approaches for stochastic process algebra using results of Kurtz [166]. Benaïm and Le Boudec [30] have also considered convergence of the underlying stochastic processes to the ODE solution in the steady-state regime, an issue which remains unresolved in the case of the ODE analyses for CTMCs described earlier. In fact, this issue is considered in Chapter 6 of this thesis, where we utilise similar ideas from stochastic approximation algorithms [29] to obtain a stationary convergence theorem in the continuous-time case.

The differential-equation approximation techniques of Benaïm and Le Boudec [30] have also gone a step further. They have noted that a generic result termed *propagation of chaos* [240] also holds in their framework. This means essentially that convergence to the mean field limit implies further that the states of any finite number of interacting objects are asymptotically independent. Such a result should also hold in the case of stochastic process algebra models although this has not been explicitly considered in the literature. Finally, Le Boudec et al. [179] describe how the mean-field approximation can also be used to focus on one individual and perform a so-called *fast simulation*, approximating the effect of the rest of the model in terms of the mean field. We mention briefly that the same author, along with others, has also considered spatial extensions of mean-field models [68] and mean field applied to Markov decision processes [107, 242], which are not considered in this thesis.

Bobbio et al. [33], Gribaudo [121], Piazzolla and Gribaudo [201] have adapted the mean-field framework of Le Boudec et al. [179] to the continuous-time case (although they do not prove a convergence result in the steady-state regime). In Bobbio et al. [33], this is used to investigate various models from queueing theory which do not always have continuous dynamics. Based on their empirical observations, it is suggested that the conditions in the convergence theorems of, for example, Kurtz [166], Le Boudec et al. [179] might feasibly be weakened to accommodate more general dynamics. In the context of the *Markovian agents* formalism [64, 121], spatial extensions to the mean-field approach have again been considered by these authors, which has been applied to the modelling of fire [67] and disaster [66] propagation, the flow of vehicles in a road tunnel [65] and wireless sensor networks [56, 122].

Bakhshi et al. [17, 18, 19] have considered discrete-time mean-field models for the analysis of so-called *gossip protocols* (see also Bakhshi [16]). This work appears to utilise the same

general framework of Le Boudec et al. [179]. Altman et al. [6] combine the mean-field approach with a branching-process approximation in their model of peer-to-peer networks in order to capture tractably properties which are not accessible through the mean-field limit alone. Finally, Gast and Gaujal [104, 105] have also considered the mean-field limiting behaviour of Markov decision processes, and very recently, Gast and Gaujal [106] have considered mean-field limits which are *differential inclusions* [4] as opposed to differential equations. These may arise when considering models with discontinuous dynamics — perhaps the result of indicator functions due to particular boundary conditions, or due to queueing behaviour present in the original model.

2.4.4 Heavy-traffic approximations for queueing networks

The approaches discussed in the preceding sections can be thought of as quite closely related to the so-called *heavy-traffic approximations* found in the queueing network literature [e.g. 34, 255]. Essentially, these approaches are based on similar fundamentals. As the number of customers in a queueing network increases, or more commonly in the queueing-network context, as the traffic intensity approaches one, the discrete-state queueing model can often be approximated by a more tractable limit process such as a differential equation or stochastic differential equation. In a sense, the ODE-based approaches for stochastic process algebra and mean-field techniques can be thought of as adapting this kind of approach to more general types of models other than just queueing networks.

It turns out in fact that the presence of queues in a model often introduces particular threshold dynamics and boundary conditions which result in distinctly discontinuous behaviour — for example, the fact that the queue length suddenly stops decreasing as soon as it reaches zero. This often demands detailed and somewhat ad-hoc technical considerations when proving a heavy-traffic limit for a particular queueing system. Much of the current research effort in this area is directed at multi-class queueing networks [e.g. 69, 170], which are particularly challenging because the very definition of heavy traffic is not even clear here [255].

This is all in quite marked contrast to the approaches discussed earlier. Both the approaches for stochastic process algebra and the mean-field method can usually be described in high levels of generality without recourse to specific ad-hoc technical arguments on a model-by-model basis. This is true even when multi-class models are considered. It would appear that this is largely attributable to the fact that most stochastic process algebra or mean-field models do not directly incorporate queues. Indeed, they usually utilise a more decentralised synchronisation approach. This is in keeping with the less centralised nature of many of the massively-parallel systems of recent interest to performance engineers.

2.4.5 Fluid/hybrid stochastic Petri nets and fluid queues

Fluid stochastic Petri nets (FSPNs) [75, 149, 252] and *fluid queueing models* [47, 99, 206] extend traditional discrete Petri net and queueing models by incorporating fluid or continuous elements directly into the model as first-class modelling components. In this way, individual *traces* of continuous components of the underlying stochastic process will satisfy differential equations which may depend on discrete elements of the model and vice-versa. This differs from the

approaches discussed above which offer continuous quantities (often computed as solutions to systems of differential equations) as approximations to stochastic measures of a fully discrete model.

Often FSPNs and fluid queueing models are used to directly capture model features which are viewed as continuous in the real world such as the level of water in a dam [191]. On the other hand, fluid levels are also often used to approximate large discrete buffer sizes in performance models of computer and communication systems in order to ameliorate the state-space explosion problem. This can in some sense be viewed as an alternative approach to addressing this problem when compared to those discussed above. However, the fundamental difference is that no formal attempt appears to be made to relate an FSPN or fluid queueing model with continuous elements back to a fully discrete model to which it is postulated as an approximation. Most importantly, no attempt appears to be made to investigate or quantify the quality of this implied approximation.

Furthermore, this direct combination of discrete and continuous components in a stochastic model can lead to rather complex systems of partial differential equations, for which numerical solution can become a computationally infeasible problem [75, 253]. Simulation then becomes the only tractable approach to analysing such models.

2.4.6 Continuous stochastic Petri nets

Continuous stochastic Petri nets (CSPNs) [86, 223, 224] are an alternative continuous approach to approximating traditional discrete stochastic Petri net models. They differ from FSPNs in that none of the Petri net places are designated as fluid places per se, rather the intention appears similar to that of the ODE-based approximation of PEPA or mean-field analysis in that continuous quantities are constructed as approximations to the original fully discrete model.

In fact, two different ways of constructing this fluid approximation are defined for CSPNs: *finite server semantics* and *infinite server semantics* [86, 223] which differ in how the rate of token flow between places is defined. Presumably, these notions are intended to correspond with the equivalent options for multiple enabling policies defined for standard discrete stochastic Petri nets [20]. In both cases, a piecewise linear differential equation defines the continuous approximation.

In terms of relating the continuous approximation to the discrete model, Júlvez et al. [156] has given a branch and bound algorithm for determining steady-state transition throughput bounds on the continuous approximation (under infinite server semantics). Specifically, the optimisation problem they construct seeks the minimal and maximal throughput obtained at a reachable (given an initial marking) fixed point of the governing system of differential equations. Asymptotic stability, that is, that the CPN will actually converge asymptotically to one of these fixed points does not appear to be considered. For a class of CPNs, the upper bound obtained in this manner is shown by Júlvez et al. [156] to coincide with a bound on the same quantity in the analogous discrete model. The bound in the discrete case was shown using techniques employing Little's Law by Campos and Silva [60]. The relationship between continuous approximation techniques and mean results derived from Little's Law has also been considered in the analysis of PEPA models by Thomas [244] and Zhao and Thomas [257].

Chapter 3

Moment approximations by differential equations for grouped PEPA

Key contributions
The grouped PEPA framework
Qualitative prediction of approximation accuracy
Higher-order moment ODE approximations

3.1 Introduction

In this chapter, we introduce the *grouped PEPA (GPEPA)* stochastic process algebra. GPEPA is a simple syntactic extension of PEPA [146] (see Section 2.3 for a summary of PEPA). The purpose of the GPEPA extension is to facilitate a clearer presentation of techniques related to differential equation-based approximation of PEPA models. The main development of this chapter is to extend first-order ODE approaches to capture key higher-order quantities such as variance in a stochastic process algebra setting. The key benefits of this generalisation are at least three-fold:

1. Variance information allows us to assess the validity of a mean prediction;
2. Higher-moment information can be used directly (see Chapter 4) to give bounds on particular kinds of passage-time distributions;
3. Rapid calculation of performance measures makes scalability analysis feasible, such as parameter sweeping and optimisation.

Indeed, this development is not restricted to PEPA-based formalisms but could be adapted fairly easily to, for example, mean-field performance models [18, 19, 30, 33, 201, 259] or continuous stochastic Petri net models [86, 224]. In addition to being extremely useful in itself, the availability of higher-order moments will also be exploited directly in Chapter 4 for the analysis of certain types of passage-time distribution.

Furthermore, we identify key qualitative predictors of the accuracy of the approximation which we term *switch points*. We show that, in general, these techniques can offer a very good approximation to quantities of interest; and furthermore, that it is largely possible to predict accurately the intervals of time over which the error is likely to be largest. The notion of a switch point is not restricted only to PEPA but could also be considered in the context of continuous stochastic Petri nets or many-server queueing networks, where the non-linear dynamics are generally also driven by minimum functions.

In this chapter we will occasionally refer the reader to Hayden and Bradley [139] for more detailed discussions. We now give a summary of the key contributions of the work presented in this chapter which should also serve as a map of the chapter's structure.

Grouped PEPA framework In Section 3.2, we develop the grouped PEPA stochastic process algebra which provides an unambiguous framework in which to define ODE-based approximations for PEPA models. There are two advantages to using the GPEPA extension: firstly, it serves to identify precisely the class of models to which this kind of analysis can be applied; and secondly, it greatly extends this class of models beyond the original work of Hillston [147] and subsequent work of others [91, 147, 250].

Qualitative prediction of approximation accuracy In Section 3.3, we define the first-moment ODE approximation in the context of the GPEPA framework and in Section 3.3.1, we present an approach to deriving this approximation directly from the underlying CTMC. This approach highlights clearly the primary source of the approximation error for GPEPA models. We are able to define a qualitative predictor of the approximation accuracy built around the notion of *switch points* and in Section 3.3.2, we show that it can be used to accurately predict intervals of time when the approximation may be expected to be at its least accurate. Although this result is fairly specific to GPEPA, it could also be directly applied to the context of continuous stochastic Petri net models which utilise a similar synchronisation dynamic. Furthermore, we note that Massink et al. [183, 184] have adapted our ideas to reason qualitatively about the error in a first-moment ODE-based analysis of emergent crowd behaviour.

Higher-order moment ODE approximations In Section 3.4, we define systems of differential equations which approximate arbitrary joint moments of component counts. This significantly and directly generalises the existing work [91, 147, 250] on ODE-based approximation of PEPA models. As mentioned briefly above, this extension of the first-order approximation could also be applied to other computer performance modelling formalisms such as mean-field and continuous stochastic Petri net based approaches. Furthermore, in Section 3.4.4, we see that switch points remain accurate predictors of approximation accuracy for higher-order moment approximations.

3.2 Grouped PEPA models

As discussed, the techniques of fluid analysis introduced are suited naturally to a particular class of PEPA model, consisting of cooperating groups of components. In order to allow such groups to be identified in the syntax explicitly we present a natural and conservative augmentation of the PEPA grammar, instances of which we call *grouped PEPA models*.

Our framework makes a clear distinction between process algebra terms which are being subject to the ODE approximation (which we term *fluid components*) and the process algebra syntax which specifies how fluid components interact with each other. Existing approaches for PEPA [91, 147, 250] implicitly include the restriction that the ODE approximation may only be applied to cooperating groups of sequential components. Our approach removes this restriction and allows the ODE approximation to be applied to cooperating groups which may themselves contain any type of PEPA component, extending the class of models to which such techniques may be applied without first having to compile general components to equivalent sequential components, an approach adopted by Clark et al. [81]. In Chapter 5, we will see that this development is particularly useful when using process algebra fragments as *stochastic probes* [8] which cooperate with the process algebra terms to be measured, resulting in non-sequential fluid components.

The GPEPA framework further extends the original work of Hillston [147] in that our techniques can be applied to models with active cooperation between components with differing rates, arbitrary cooperation sets between groups of components, and the presence of more than one component enabling the same action type on either side of a cooperation. These particular limitations were originally overcome by Hayden [129] and have since then also been dealt with in work by Tribastone et al. [250] and Ding [91].

3.2.1 Syntax

We will maintain the standard PEPA component definitions from Section 2.3:

$$\begin{aligned} S &::= (\alpha, r).S \;\mid\; S + S \;\mid\; C_S \\ P &::= P \underset{L}{\bowtie} P \;\mid\; P/L \;\mid\; S \;\mid\; C_P \end{aligned} \tag{3.1}$$

We wish to define a new type of component that represents explicitly a purely concurrent group of standard PEPA components. To this end we define a *component group* D:

$$D ::= D \between D \;\mid\; P \tag{3.2}$$

We refer to the standard PEPA components P (Section 2.3) within a component group as *fluid components*. The fluid components are so-called because it is these components whose state will be tracked explicitly by an approximating system of differential equations. The additional combinator $\between$ will be defined to have the same meaning as $\|$. Its purpose is simply to distinguish between parallel cooperation *within* fluid components and the parallel cooperation *between* the fluid components constituting the component group. The reason for this extra hierarchy is to specify syntactically the level at which the ODE-based analysis is to be performed, that is, at the level of component groups.

Cooperation within components groups is not considered since such models do not have an obvious fluid interpretation. Specifically, an attempt to apply the ideas of Section 2.4.1 to derive ODEs approximating such models will, in general, result in differential equations with jumps in their right-hand sides. For example, consider the simple PEPA model $A(n)$, defined

by:

$$\begin{aligned}\boldsymbol{A}_0 &\stackrel{def}{=} (a,\lambda).\boldsymbol{A}_1\\ \boldsymbol{A}_1 &\stackrel{def}{=} (b,\mu).\boldsymbol{A}_0\\ \boldsymbol{A}(n) &\stackrel{def}{=} \underbrace{\boldsymbol{A}_0 \underset{\{a\}}{\bowtie} \boldsymbol{A}_0 \underset{\{a\}}{\bowtie} \ldots \underset{\{a\}}{\bowtie} \boldsymbol{A}_0}_{n}\end{aligned}$$

Deriving ODEs directly from its natural state-space aggregation, as in the case of the model $\boldsymbol{PR}(n,m)$, yields:

$$\begin{aligned}\dot{\boldsymbol{A}}_0(t) &= \mu \boldsymbol{A}_1(t) - \lambda n \mathbf{I}'(\boldsymbol{A}_1(t))\\ \dot{\boldsymbol{A}}_1(t) &= -\mu \boldsymbol{A}_1(t) + \lambda n \mathbf{I}'(\boldsymbol{A}_1(t))\end{aligned}$$

where $\mathbf{I}' := 1 - \mathbf{I}$ is the complement of the indicator function:

$$\mathbf{I}(x) := \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \end{cases}$$

The discontinuous indicator functions follow from the fact that the model can only perform the a-action when all of the members of the group are in the state $\boldsymbol{A}_0$, because of the internal cooperation. Due to their presence, these equations cannot be interpreted as ordinary differential equations, even if only single-sided derivatives are demanded. For very similar reasons, we restrict all passive actions to be synchronised in a final (outermost) fluid component, so as not to allow passive cooperation across component groups. Formally, this is stated by the requirement that in Equation 3.2, if $P' \in ds(P)$, then there is no $(\alpha, r) \in \mathcal{A}ct(P')$ with $r \notin \mathbb{R}_+$.[1]

The next definition allows the arbitrary combination of component groups into a *grouped PEPA model* M:

$$M ::= M \underset{L}{\bowtie} M \;\mid\; Y\{D\} \tag{3.3}$$

where $L \subseteq \mathcal{A}_t \setminus \{\tau\}$ is a cooperation set.[2] A grouped PEPA model consists of arbitrarily cooperating component groups (instances of $Y\{D\}$, called *labelled component groups*) and each labelled component group in a given model is assumed to have a distinct label Y. From now on, we will also refer to grouped PEPA models as *GPEPA* models.

Consider again the standard PEPA model $\boldsymbol{PR}(n,m)$ of Section 2.3.4. A natural representation of this standard PEPA model as a grouped PEPA model would be:

$$\boldsymbol{PR}(n,m) \stackrel{def}{=} \mathbf{P}\{\boldsymbol{Proc}_0[n]\} \underset{\{task_a\}}{\bowtie} \mathbf{R}\{\boldsymbol{Res}_0[m]\}$$

where for grouped PEPA models, $C[k] := \underbrace{(C \between \ldots \between C)}_{k}$. So the fluid components are the $\boldsymbol{Proc}_i$ and $\boldsymbol{Res}_i$, that is, the fluid approximation will consist of four coupled differential equations

[1] Although outside the scope of this thesis, Hayden and Bradley [140] have considered an approach which allows ODE-based analysis to be performed on GPEPA models with passive cooperation between component groups.

[2] Note that for the sake of brevity and clarity we do not explicitly consider action hiding at the level of grouped PEPA models in this thesis. However, it is discussed in detail in Hayden and Bradley [139].

counting the number of each of these fluid components active in the model. Alternatively, for example, the following GPEPA model representation of the same standard PEPA model is also possible (assuming n is a multiple of 2):

$$\mathbf{P}\{(\boldsymbol{Proc}_0 \parallel \boldsymbol{Proc}_0)[n/2]\} \underset{\{task_a\}}{\bowtie} \mathbf{R}\{\boldsymbol{Res}_0[m]\}$$

In this case, we consider each pair of processors as a single fluid component. So there will be six coupled differential equations since $|ds(\boldsymbol{Proc}_0 \parallel \boldsymbol{Proc}_0)| = 4$.

3.2.2 Semantics

A GPEPA model should be thought of simply as the corresponding standard PEPA model, but with additional annotations (the group labellings), which specify unambiguously the level at which the fluid analysis should be performed. We therefore define the operational semantics of a GPEPA model to be essentially identical to that of the equivalent standard PEPA model which is obtained syntactically by removing the group labels and replacing the $\wr$ combinators with $\parallel$. We call this operation *flattening* and a formal *flattening function* is given below.

Definition 3.2.1 (GPEPA :: Flattening function)**.** *For any GPEPA model G, the corresponding standard PEPA model $\mathcal{F}(G)$ can be recovered from the grouped model as follows.*

$$\mathcal{F}(M_1 \underset{L}{\bowtie} M_2) := \mathcal{F}(M_1) \underset{L}{\bowtie} \mathcal{F}(M_2)$$
$$\mathcal{F}(Y\{D\}) := \mathcal{F}'(D)$$

where for component groups:

$$\mathcal{F}'(D_1 \wr D_2) := \mathcal{F}'(D_1) \parallel \mathcal{F}'(D_2)$$
$$\mathcal{F}'(P) := P$$

We may then formally define operational semantics for GPEPA models by stating that for any two GPEPA models G_1 and G_2 and timed action type $\alpha \in \mathcal{A}_t$, $G_1 \xrightarrow{(\alpha,r)} G_2$ if and only if $\mathcal{F}(G_1) \xrightarrow{(\alpha,r)} \mathcal{F}(G_2)$, counting also the multiplicity of such transitions.

Mirroring standard PEPA, we may then also naturally define the set of derivative states of a GPEPA model $ds(G)$ as the set of all GPEPA components reachable from it by timed transitions.[3] Also, we extend naturally to a GPEPA model G, the definitions of apparent rate $r_\alpha(G) := r_\alpha(\mathcal{F}(G))$ for $\alpha \in \mathcal{A}_t$, timed action set $\mathcal{A}_t(G) := \mathcal{A}_t(\mathcal{F}(G))$ and activity set $\mathcal{A}ct(G) := \mathcal{A}ct(\mathcal{F}(G))$. Finally, by naturally constructing a labelled and directed multigraph as we did for a PEPA model in Section 2.3.3, we may define the underlying CTMC associated to a GPEPA model.

Some properties of GPEPA models required for the following sections are now given in Table 3.2, illustrated by example. More formal definitions can be found in Hayden and Bradley [139].

[3] We overload the syntax when equivalent definitions are effectively the same for standard PEPA models and GPEPA models.

$\mathcal{G}(G)$	The set of all component group labels in the GPEPA model G, e.g. $\mathcal{G}(\boldsymbol{PR}(n,m)) = \{\mathbf{P}, \mathbf{R}\}$.
$\mathcal{B}(G,H)$	The union of all derivative states of all fluid components in the component group of G which has group label H, e.g. $\mathcal{B}(\boldsymbol{PR}(n,m), \mathbf{P}) = \{\boldsymbol{Proc}_0, \boldsymbol{Proc}_1\}$.
$\mathcal{B}(G)$	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. $\mathcal{B}(\boldsymbol{PR}(n,m)) = \{(\mathbf{P}, \boldsymbol{Proc}_0), (\mathbf{P}, \boldsymbol{Proc}_1), (\mathbf{R}, \boldsymbol{Res}_0), (\mathbf{R}, \boldsymbol{Res}_1)\}$.
$\mathcal{N}(G)$	The number of different types of derivative states of fluid components in each group of G, equal to the number of differential equations required to specify the fluid analysis. Thus $\mathcal{N}(G) = \lvert\mathcal{B}(G)\rvert$, e.g. $\mathcal{N}(\boldsymbol{PR}(n,m)) = 4$.
$\mathcal{S}(G,H)$	The size of the component group with label H. That is, the number of fluid components in the group, e.g. $\mathcal{S}(\boldsymbol{PR}(n,m), \mathbf{P}) = n$.
$\mathcal{C}(G,H,Q)$	The integer count of fluid components in state Q in the component group of G which has group label H, e.g. $\mathcal{C}(\boldsymbol{PR}(n,m), \mathbf{P}, \boldsymbol{Proc}_0) = n$.
$\mathcal{S}(G)$	$\mathcal{S}(G) := \sum_{H \in \mathcal{G}(G)} \mathcal{S}(G,H)$, i.e. the total size of all component groups in G, e.g. $\mathcal{S}(\boldsymbol{PR}(n,m)) = n + m$.

Table 3.1: Key GPEPA definitions.

3.2.3 Running example: client–server model

In this section, we will introduce a fairly simple, but non-trivial, GPEPA model which will serve as a running example throughout this chapter and which will also be used in some of the later chapters. In the GPEPA model $\boldsymbol{CS}(n,m)$ given in Figure 3.1, we have a population of n clients and a population of m servers. The system uses a 2-stage fetch mechanism: a client requests data from the pool of servers; one of the servers receives the request and another server may then fetch the data for the client. At any stage, a server in the pool may fail. Clients may also timeout when waiting for data after their initial request.

3.2.4 Aggregation of GPEPA models

We now proceed to show how a GPEPA model's state space may be aggregated by combining states where the component counts agree. This will be achieved by defining the relation *groupwise equivalence* on the derivative states of grouped PEPA models.

Definition 3.2.2 (GPEPA :: Groupwise equivalence)**.** *Let G be a GPEPA model. Define $G_1, G_2 \in ds(G)$ to be groupwise equivalent, denoted by $G_1 \simeq G_2$, if and only if for all $(H,P) \in \mathcal{B}(G)$, we have $\mathcal{C}(G_1,H,P) = \mathcal{C}(G_2,H,P)$.*

Theorem 3.2.3. *$\simeq$ is an equivalence relation, i.e. it induces a partition $ds(G)/\simeq$ on the state space, $ds(G)$, of a GPEPA model G. Furthermore, this partition is (ordinarily) lumpable (Section 2.2.3).*

Proof. That $\simeq$ is an equivalence relation is clear from its definition. To see lumpability, let $G_1 \simeq G_2$ for G_1 and $G_2 \in ds(G)$. Let $\mathsf{G} \in ds(G)/\simeq$. Lumpability then follows from the fact

$$\begin{aligned}
\boldsymbol{Client} &\stackrel{def}{=} (request, r_r).\boldsymbol{Client_{waiting}} \\
\boldsymbol{Client_{waiting}} &\stackrel{def}{=} (data, r_d).\boldsymbol{Client_{think}} + (timeout, r_{tmt}).\boldsymbol{Client} \\
\boldsymbol{Client_{think}} &\stackrel{def}{=} (think, r_t).\boldsymbol{Client} \\
\boldsymbol{Server} &\stackrel{def}{=} (request, r_r).\boldsymbol{Server_{get}} + (break, r_b).\boldsymbol{Server_{broken}} \\
\boldsymbol{Server_{get}} &\stackrel{def}{=} (data, r_d).\boldsymbol{Server} + (break, r_b).\boldsymbol{Server_{broken}} \\
\boldsymbol{Server_{broken}} &\stackrel{def}{=} (reset, r_{rst}).\boldsymbol{Server} \\
\\
\boldsymbol{CS}(n, m) &\stackrel{def}{=} \mathbf{Clients}\{\boldsymbol{Client}[n]\} \underset{\{request, data\}}{\bowtie} \mathbf{Servers}\{\boldsymbol{Server}[m]\}
\end{aligned}$$

Figure 3.1: GPEPA description of the simple client–server model $\boldsymbol{CS}(n, m)$.

that:

$$\sum_{G' \in \mathsf{G}} q(G_1, G', \alpha) = \sum_{G' \in \mathsf{G}} q(G_2, G', \alpha)$$

for all $\alpha \in \mathcal{A}_t$, where $q(G, G', \alpha) := \sum_{G \xrightarrow{(\alpha, r)} G'} r$ is the sum of the rates of all α-transitions between two GPEPA derivative states. This follows immediately from the operational semantics of GPEPA (Figure 2.3) since transition rates are independent of the order of fluid components within component groups. □

Therefore $\simeq$ induces a lumpable partition on the state space of the underlying CTMC of a GPEPA model. We refer to this as the *underlying aggregated CTMC* of the model. 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 $\mathsf{N} \in \mathcal{B}(G) \to \mathbb{Z}_+$.[4] This function counts the number of fluid components in each derivative state currently active in a given component group.

For example, in the case of $G = \boldsymbol{CS}(n, m)$, one equivalence class $\mathsf{G} \in ds(G)/\simeq$ is:

$$\begin{aligned}
\mathsf{G} = \{&\mathbf{Clients}\{\boldsymbol{Client} \wr \boldsymbol{Client_{think}}[n-1]\} \underset{L}{\bowtie} \mathbf{Servers}\{\boldsymbol{Server}[m]\}, \\
&\mathbf{Clients}\{\boldsymbol{Client_{think}} \wr \boldsymbol{Client} \wr \boldsymbol{Client_{think}}[n-2]\} \underset{L}{\bowtie} \mathbf{Servers}\{\boldsymbol{Server}[m]\}, \\
&\mathbf{Clients}\{\boldsymbol{Client_{think}}[2] \wr \boldsymbol{Client} \wr \boldsymbol{Client_{think}}[n-3]\} \underset{L}{\bowtie} \mathbf{Servers}\{\boldsymbol{Server}[m]\}, \\
&\ldots, \\
&\mathbf{Clients}\{\boldsymbol{Client_{think}}[n-1] \wr \boldsymbol{Client}\} \underset{L}{\bowtie} \mathbf{Servers}\{\boldsymbol{Server}[m]\}\}
\end{aligned}$$

This is the case of one $\boldsymbol{Client}$ component, $(n-1) \times \boldsymbol{Client_{think}}$ components and no $\boldsymbol{Client_{waiting}}$ components; and $m \times \boldsymbol{Server}$ components, no $\boldsymbol{Server_{get}}$ or $\boldsymbol{Server_{broken}}$ components. It is therefore represented by the function $\mathsf{N} \in \mathcal{B}(\boldsymbol{CS}(n, m)) \to \mathbb{Z}_+$, defined by $\mathsf{N}(\mathbf{Clients}, \boldsymbol{Client}) = 1$, $\mathsf{N}(\mathbf{Clients}, \boldsymbol{Client_{waiting}}) = 0$, $\mathsf{N}(\mathbf{Clients}, \boldsymbol{Client_{think}}) = n - 1$, $\mathsf{N}(\mathbf{Servers}, \boldsymbol{Server}) = m$ and $\mathsf{N}(\mathbf{Servers}, \boldsymbol{Server_{get}}) = \mathsf{N}(\mathbf{Servers}, \boldsymbol{Server_{broken}}) = 0$.

[4] Conversely, note that not all such functions specify valid states in the underlying aggregated CTMC. For example, such a function may specify a total number of fluid components in a component group that exceeds the component group's size, or may specify an otherwise unreachable CTMC state.

3.2.5 Evolution rates of GPEPA models

The purpose of this section is to define the *component rate function* for a GPEPA model G, which calculates the total rate at which a fluid component evolves, taking into account any interactions with the rest of the model. Specifically, $\mathcal{R}_\alpha(G, \mathsf{N}, H, P)$ defines the component rate of P within a component group H completing a timed action $\alpha \in \mathcal{A}_t$, in the aggregate state specified by $\mathsf{N} \in \mathcal{B}(G) \to \mathbb{R}_+$. We allow also non-integer 'component counts' since the concepts we define will also be used later to define the real-valued ODE approximations. The component rate function is needed to describe the rate of evolution of a component from one derivative state to the next in the construction of the ODEs which will define the ODE approximation.

We will also need to define a *derivative weighting function* and a version of the *apparent rate function* defined in terms of component counts $\mathsf{N} \in \mathcal{B}(G) \to \mathbb{R}_+$. Table 3.2 summarises the notation used for these functions.

$\mathcal{R}_\alpha(G, \mathsf{N}, H, P)$	The **component rate function** measures the local rate at which fluid components in state P in group H perform α-actions in the context of the cooperation within the wider grouped PEPA model G in aggregate state given by N.
$p_\alpha(P, Q)$	The **derivative weighting function** measures the probability that a fluid component P evolves to state Q in one α-transition.
$r_\alpha(G, \mathsf{N})$	The **apparent rate function** measures the total rate of α-actions enabled by G in the aggregate state given by N.

Table 3.2: GPEPA rate functions.

Definition 3.2.4 (GPEPA :: Component rate function)**.** *Let G be a GPEPA model. For $(H, P) \in \mathcal{B}(G)$, timed action type $\alpha \in \mathcal{A}_t$ and $\mathsf{N} \in \mathcal{B}(G) \to \mathbb{R}_+$, the component rate function is $\mathcal{R}_\alpha(G, \mathsf{N}, H, P)$, defined as follows.*

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

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

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

The component rate function gives the overall aggregate rate at which a given fluid component within a GPEPA model performs a certain action type. We are ultimately interested however in how that rate is shared between enabled transitions into different derivative states. To this end, we define the *derivative weighting function* which calculates the probability that given that a fluid component performs an action, it transits to another specified fluid component state.

Definition 3.2.5 (GPEPA :: Derivative weighting function)**.** *Let P and Q be fluid (standard PEPA) components and let $\alpha \in \mathcal{A}_t$. Then $p_\alpha(P,Q) := \frac{1}{r_\alpha(P)} \sum_{P \xrightarrow{(\alpha,\lambda)} Q} \lambda$. This is defined to be zero when $r_\alpha(P) = 0$.*

Finally, in order to support the preceding two definitions, we require a version of the apparent rate function defined on GPEPA models explicitly in terms of their component counts $\mathsf{N} \in \mathcal{B}(G) \to \mathbb{R}_+$. It is straightforward to see that this version matches the definition $r_\alpha(G) := r_\alpha(\mathcal{F}(G))$ given earlier in Section 3.2.2.

Definition 3.2.6 (GPEPA :: Apparent rate)**.** *Let G be a GPEPA model. Let $\alpha \in \mathcal{A}_t$ be a timed action type and $\mathsf{N} \in \mathcal{B}(G) \to \mathbb{R}_+$. Then the apparent rate is $r_\alpha(G, \mathsf{N})$, defined as follows.*

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

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

We now wish to relate formally the component rate function and the derivative weighting function to the underlying aggregated CTMC induced by the equivalence relation $\simeq$. This is achieved by characterising the outgoing transitions from a given state in terms of these functions in the following theorem.

Theorem 3.2.7. *Let G be a GPEPA model. Let $\mathsf{N} \in \mathcal{B}(G) \to \mathbb{Z}_+$ represent a state of the aggregated CTMC of G. Consider the component group with label $H \in \mathcal{G}(G)$ and let $\alpha \in \mathcal{A}_t$ be a timed action type. For each fluid component $P \in \mathcal{B}(G, H)$, all α-transitions from N to some other aggregated state $\mathsf{N}' \in \mathcal{B}(G) \to \mathbb{Z}_+$ where $\mathsf{N}'(H, P) \neq \mathsf{N}(H, P)$ are such that exactly one of the following holds:*

1. *$\mathsf{N}'(H, P) = \mathsf{N}(H, P) - 1$, and furthermore, the sum of the rates of all α-transitions to states N' for which this holds is:*

$$\sum_{\substack{Q \in \mathcal{B}(G,H) \\ Q \neq P}} p_\alpha(P, Q)\, \mathcal{R}_\alpha(G, \mathsf{N}, H, P)$$

   or

2. *$\mathsf{N}'(H, P) = \mathsf{N}(H, P) + 1$ and furthermore, the sum of the rates of all α-transitions to states N' for which this holds is:*

$$\sum_{\substack{Q \in \mathcal{B}(G,H) \\ Q \neq P}} p_\alpha(Q, P)\, \mathcal{R}_\alpha(G, \mathsf{N}, H, Q)$$

Also in either case, if there is no such α-transition, the rates given above are zero.

Proof. See Hayden [131, Theorem 2.15]. □

By way of example, consider the aggregate state represented by the function $\mathsf{N} \in \mathcal{B}(\boldsymbol{CS}(20,10)) \to \mathbb{Z}_+$, defined by $\mathsf{N}(\mathbf{Clients}, \boldsymbol{Client}) = 5$, $\mathsf{N}(\mathbf{Clients}, \boldsymbol{Client_{waiting}}) = 5$, $\mathsf{N}(\mathbf{Clients}, \boldsymbol{Client_{think}}) = 10$, $\mathsf{N}(\mathbf{Servers}, \boldsymbol{Server}) = 4$, $\mathsf{N}(\mathbf{Servers}, \boldsymbol{Server_{get}}) = 3$, and $\mathsf{N}(\mathbf{Servers}, \boldsymbol{Server_{broken}}) = 3$. In the above theorem, let $(H, P) = (\mathbf{Clients}, \boldsymbol{Client_{think}})$ and $\alpha = \mathit{think}$, then the quantity of Item 1 in the theorem is:

$$\sum_{\substack{Q \in \mathcal{B}(\boldsymbol{CS}(n,m), \mathbf{Clients}) \\ Q \neq \boldsymbol{Client_{think}}}} p_{\mathit{think}}(\boldsymbol{Client_{think}}, Q)\, \mathcal{R}_{\mathit{think}}(\boldsymbol{CS}(n,m), \mathsf{N}, \mathbf{Clients}, \boldsymbol{Client_{think}}) = 10 \times r_t$$

This is the sum of the rate of all transitions which *decrease* the count of $\boldsymbol{Client_{think}}$ components in this state. The quantity of Item 2 in the theorem is:

$$\sum_{\substack{Q \in \mathcal{B}(\boldsymbol{CS}(n,m), \mathbf{Clients}) \\ Q \neq \boldsymbol{Client_{think}}}} p_{\mathit{think}}(Q, \boldsymbol{Client_{think}})\, \mathcal{R}_{\mathit{think}}(\boldsymbol{CS}(n,m), \mathsf{N}, \mathbf{Clients}, Q) = \min(5,3) r_t = 3 \times r_t$$

This is the sum of the rate of all transitions which *increase* the count of $\boldsymbol{Client_{think}}$ components in this state.

3.3 Approximation of first moments

We now proceed to show how a system of coupled ODEs may be derived from a GPEPA model. These ODEs describe a continuous approximation to the time evolution of the numbers of each type of fluid component within the GPEPA model.

Consider a GPEPA model G. Let $(H, P) \in \mathcal{B}(G)$ represent a particular fluid component derivative state. We will now introduce functions $v_{H,P}(t)$ as deterministic, continuous approximations to the integer-valued stochastic processes $N_{H,P}(t) := \mathsf{N}_t(H, P)$ where $\mathsf{N}_t \in \mathcal{B}(G) \to \mathbb{Z}_+$ represents the state of the underlying aggregated CTMC at time $t \in \mathbb{R}_+$. In Section 3.3.1, we will then discuss how the functions $v_{H,P}(t)$ can be interpreted as approximations to the first moments of the component counts $\mathbb{E}[N_{H,P}(t)]$. The definition of $v_{H,P}(t)$ is by means of the system of coupled, first-order ODEs which follow.

Definition 3.3.1 (GPEPA :: First-moment ODEs)**.** *Let G be a GPEPA model. We define the evolution of the $v_{H,P}(t)$ over time for $(H, P) \in \mathcal{B}(G)$ by the system of first-order coupled ODEs:*

$$\dot{v}_{H,P}(t) = \sum_{\alpha \in \mathcal{A}_t} \left(-\mathcal{R}_\alpha(G, \mathsf{V}_t, H, P) + \sum_{Q \in \mathcal{B}(G,H)} p_\alpha(Q, P)\, \mathcal{R}_\alpha(G, \mathsf{V}_t, H, Q) \right)$$
$$\textit{for all } (H, P) \in \mathcal{B}(G)$$

where for $t \in \mathbb{R}_+$, $\mathsf{V}_t \in \mathcal{B}(G) \to \mathbb{R}$ is given by $\mathsf{V}_t(H, P) := v_{H,P}(t)$ for all $(H, P) \in \mathcal{B}(G)$. The initial conditions $\mathsf{V}_0 \in \mathcal{B}(G) \to \mathbb{R}$ for this system of ODEs are those naturally defined by the initial state of G. That is, $\mathsf{V}_0(H, P) := \mathcal{C}(G, H, P)$.

These ODEs can be intuitively justified by Theorem 3.2.7 if we consider the rate of all outgoing α-transitions which *increase* the number of fluid components in derivative state P minus the

rate of all outgoing α-transitions which *decrease* the number of fluid components in derivative state P.

Note that for non-negative initial conditions, it is immediate from the definition of the ODEs that for any solution, $\dot{v}_{H,P}(t) \geq -v_{H,P}(t)$ and thus, $v_{H,P}(t) \geq 0$ for all $t \in \mathbb{R}_+$. Furthermore $v_{H,P}(t) \leq \mathcal{S}(G,H)$ for all $t \in \mathbb{R}_+$ since for all $H \in \mathcal{G}(G)$, $\sum_{P \in \mathcal{B}(G,H)} \dot{v}_{H,P}(t) = 0$ and $\mathsf{V}_0(H,P) \leq \mathcal{S}(G,H)$. That is, any solution to the system of ODEs must at least lie within the natural boundaries imposed by the model they are derived from.

We will sometimes find it useful to refer to the aggregated CTMC and the ODE approximation as vectors $\mathbf{N}(t) \in \mathbb{Z}_+^{\mathcal{N}(G)}$ and $\mathbf{v}(t) \in \mathbb{R}_+^{\mathcal{N}(G)}$, respectively by applying some ordering to the elements of $\mathcal{B}(G)$. We can then write the above system of ODEs as $\dot{\mathbf{v}}(t) = \mathbf{f}(\mathbf{v}(t))$ where $\mathbf{f}$ is a function in $\mathbb{R}_+^{\mathcal{N}(G)} \to \mathbb{R}^{\mathcal{N}(G)}$. Note that we have intentionally suppressed the dependence of $\mathbf{N}(t)$, $\mathbf{v}(t)$ and $\mathbf{f}$ on G so as not to burden the notation too far.

In Appendix A.1, we verify that $\mathbf{f}$ is a Lipschitz continuous function. By standard results on differential equations (see e.g. Miller and Michel [187]) this guarantees that the system of ODEs in Definition 3.3.1 has a unique solution.

Finally, the following straightforward result shows that any two solutions to $\dot{\mathbf{v}}(t) = \mathbf{f}(\mathbf{v}(t))$ where the initial conditions for each solution are a constant scalar multiple of the other remain scalar multiples of each other for all $t \in \mathbb{R}_+$.

Lemma 3.3.2. *The function* $\mathbf{f}$ *enjoys the following homogeneity property. For any* $\beta \in \mathbb{R}_+$, $\mathbf{f}(\beta\mathbf{x}) = \beta\mathbf{f}(\mathbf{x})$ *for all* $\mathbf{x} \in \mathbb{R}_+^{\mathcal{N}(G)}$.

Proof. This follows from the homogeneity of the apparent and component rate functions: $r_\alpha(G, \beta\mathsf{N}) = \beta r_\alpha(G,\mathsf{N})$ and $\mathcal{R}_\alpha(G,\beta\mathsf{N},H,P) = \beta\mathcal{R}_\alpha(G,\mathsf{N},H,P)$ for all $(H,P) \in \mathcal{B}(G)$ and $\mathsf{N} \in \mathcal{B}(G) \to \mathbb{R}_+$. □

If we apply Definition 3.3.1 to the client–server model of Section 3.2.3, we obtain the following system of ODEs:[5]

$$\begin{aligned}
\dot{v}_{\boldsymbol{C}}(t) &= -\min(v_{\boldsymbol{C}}(t), v_{\boldsymbol{S}}(t))r_r + v_{\boldsymbol{Cw}}(t)r_{tmt} + v_{\boldsymbol{Ct}}(t)r_t \\
\dot{v}_{\boldsymbol{Cw}}(t) &= -\min(v_{\boldsymbol{Cw}}(t), v_{\boldsymbol{Sg}}(t))r_d - v_{\boldsymbol{Cw}}(t)r_{tmt} + \min(v_{\boldsymbol{C}}(t), v_{\boldsymbol{S}}(t))r_r \\
\dot{v}_{\boldsymbol{Ct}}(t) &= -v_{\boldsymbol{Ct}}(t)r_t + \min(v_{\boldsymbol{Cw}}(t), v_{\boldsymbol{Sg}}(t))r_d \\
\dot{v}_{\boldsymbol{S}}(t) &= -\min(v_{\boldsymbol{C}}(t), v_{\boldsymbol{S}}(t))r_r - v_{\boldsymbol{S}}(t)r_b + \min(v_{\boldsymbol{Cw}}(t), v_{\boldsymbol{Sg}}(t))r_d + v_{\boldsymbol{Sb}}(t)r_{rst} \\
\dot{v}_{\boldsymbol{Sg}}(t) &= -\min(v_{\boldsymbol{Cw}}(t), v_{\boldsymbol{Sg}}(t))r_d - v_{\boldsymbol{Sg}}(t)r_b + \min(v_{\boldsymbol{C}}(t), v_{\boldsymbol{S}}(t))r_r \\
\dot{v}_{\boldsymbol{Sb}}(t) &= -v_{\boldsymbol{Sb}}(t)r_{rst} + v_{\boldsymbol{S}}(t)r_b + v_{\boldsymbol{Sg}}(t)r_b
\end{aligned} \tag{3.4}$$

These ODEs are also available in MATLAB® form from the online web-based companion to this thesis [131]. The accuracy of the approximation obtained by solving this system is discussed later in Section 3.3.2.

[5] Throughout this thesis, we will often adopt obvious shorthands for the ODE solutions and component counting processes, such as $v_{\boldsymbol{C}}(t)$ for $v_{\mathbf{Clients},\boldsymbol{Client}}(t)$ and $v_{\boldsymbol{Sb}}(t)$ for $v_{\mathbf{Servers},\boldsymbol{Server_{broken}}}(t)$; and $N_{\boldsymbol{C}}(t)$ for $N_{\mathbf{Clients},\boldsymbol{Client}}(t)$ and $N_{\boldsymbol{Sb}}(t)$ for $N_{\mathbf{Servers},\boldsymbol{Server_{broken}}}(t)$.

3.3.1 Nature of the first-moment approximation

In this section we will show how the quantity $v_{H,P}(t)$ can be seen as an approximation to the corresponding component count expectation $\mathbb{E}[N_{H,P}(t)]$. Then in Section 3.4, we will see how we may extend this argument to develop similar approximate analyses for (joint) higher-order moments of the stochastic processes $N_{H,P}(t)$.

We will proceed by characterising the action of the infinitesimal generator matrix of the underlying CTMC of G, denoted $\mathbf{A}_G$, on component counts at a time $t \in \mathbb{R}_+$. Recall from Section 2.1 how the infinitesimal generator can be interpreted as a linear operator. Let $(H,P) \in \mathcal{B}(G)$ be the fluid component of interest and define the stochastic process $R_{H,P}(t)$ as the total rate of all outgoing transitions in the CTMC state occupied at time t *which do not change the count of P-components in group H*. Using Theorem 3.2.7, we may then see that the action of the infinitesimal generator on component counts is given by:

$$\begin{aligned}
\mathbf{A}_G\, N_{H,P}(t) = &- \left(\sum_{\alpha \in \mathcal{A}_t} \sum_{\substack{Q \in \mathcal{B}(G,H) \\ Q \neq P}} p_\alpha(P,Q)\, \mathcal{R}_\alpha(G, \mathsf{N}_t, H, P) \right) N_{H,P}(t) \\
&- \left(\sum_{\alpha \in \mathcal{A}_t} \sum_{\substack{Q \in \mathcal{B}(G,H) \\ Q \neq P}} p_\alpha(Q,P)\, \mathcal{R}_\alpha(G, \mathsf{N}_t, H, Q) \right) N_{H,P}(t) \\
&- R_{H,P}(t)\, N_{H,P}(t) \\
&+ \left(\sum_{\alpha \in \mathcal{A}_t} \sum_{\substack{Q \in \mathcal{B}(G,H) \\ Q \neq P}} p_\alpha(P,Q)\, \mathcal{R}_\alpha(G, \mathsf{N}_t, H, P) \right) (N_{H,P}(t) - 1) \\
&+ \left(\sum_{\alpha \in \mathcal{A}_t} \sum_{\substack{Q \in \mathcal{B}(G,H) \\ Q \neq P}} p_\alpha(Q,P)\, \mathcal{R}_\alpha(G, \mathsf{N}_t, H, Q) \right) (N_{H,P}(t) + 1) \\
&+ R_{H,P}(t)\, N_{H,P}(t)
\end{aligned}$$

After simplification, this becomes:

$$\mathbf{A}_G\, N_{H,P}(t) = \sum_{\alpha \in \mathcal{A}_t} \left(\sum_{Q \in \mathcal{B}(G,H)} p_\alpha(Q,P)\, \mathcal{R}_\alpha(G, \mathsf{N}_t, H, Q) - \mathcal{R}_\alpha(G, \mathsf{N}_t, H, P) \right)$$

We may then apply Equation 2.1 directly, which yields the following exact differential equation for the component-count expectation $\mathbb{E}[N_{H,P}(t)]$:

$$\dot{\mathbb{E}}[N_{H,P}(t)] = \sum_{\alpha \in \mathcal{A}_t} \left(\sum_{Q \in \mathcal{B}(G,H)} p_\alpha(Q,P)\, \mathbb{E}[\mathcal{R}_\alpha(G, \mathsf{N}_t, H, Q)] - \mathbb{E}[\mathcal{R}_\alpha(G, \mathsf{N}_t, H, P)] \right) \tag{3.5}$$

This differential equation states that the component count expectations almost satisfy the differential equations of Definition 3.3.1. The problem is that Equation 3.5 contains expectations

of possibly non-linear functions in its right-hand side: specifically, expectations of component rate functions, such as $\mathbb{E}[\mathcal{R}_\alpha(G, \mathsf{N}_t, H, Q)]$. In the case that the component rate function is a linear function of the component counts, we have that:

$$\mathbb{E}[\mathcal{R}_\alpha(G, \mathsf{N}_t, H, Q)] = \mathcal{R}_\alpha(G, \mathsf{E}_t, H, Q) \tag{3.6}$$

for any fluid component Q, where $\mathsf{E}_t(H, P) := \mathbb{E}[N_{H,P}(t)]$. Then, the component count expectations satisfy the ODEs of Definition 3.3.1 exactly. However, it is easy to see from its definition (Definition 3.2.4) that the component rate function will only adopt such a simple form in the case of there being no synchronisation between component groups. In such models, components evolve entirely independently of each other and the underlying stochastic process enjoys a transient product-form solution, so is already easily tractable and not particularly interesting.

In the more general interesting case where synchronisation is allowed between component groups, Equation 3.6 becomes an approximation. More precisely, in this case the solutions to the ODEs of Definition 3.3.1 $v_{H,P}(t)$ represent an approximation to the component count expectations $\mathbb{E}[N_{H,P}(t)]$, the quality of which relies directly on the difference between the terms $\mathbb{E}[\mathcal{R}_\alpha(G, \mathsf{N}_t, H, Q)]$ and $\mathcal{R}_\alpha(G, \mathsf{E}_t, H, Q)$. For a particular class of general GPEPA models, this approximation has a form which means that it is possible to predict with certainty when it is likely to be most inaccurate. Models in this class are said to be *split-free* and are introduced and discussed in the next section.

Split-free models and switch points

The class of *split-free models* is general enough to include many massively-parallel models encountered in practice. Specifically it *excludes* models involving cooperations $M_1 \underset{L}{\bowtie} M_2$ where, for some shared timed action $\alpha \in L$, either M_1 or M_2 contains more than one type of fluid component component which enables α. That is, at least one side of the cooperation *splits* between two different types of fluid component over this action type. Models not included in this class are referred to as *splitting*. A formal definition of a split-free GPEPA model follows.

Definition 3.3.3 (GPEPA :: Split-free model)**.** *A GPEPA model G is split-free if and only if $\mathcal{Q}(G) = \mathbf{true}$, defined as follows:*

$$\begin{aligned} \mathcal{Q}(M_1 \underset{L}{\bowtie} M_2) &:= \mathcal{Q}(M_1) \wedge \mathcal{Q}(M_2) \wedge \mathcal{Q}'(M_1,\, L) \wedge \mathcal{Q}'(M_2,\, L) \\ \mathcal{Q}(Y\{D\}) &:= \mathbf{true} \end{aligned}$$

where $\mathcal{Q}'(M,\, L) = \mathbf{true}$ if and only if for all $\alpha \in L$, there do not exist distinct $(H_1, P_1), (H_2, P_2) \in \mathcal{B}(M)$ such that $(\alpha, \cdot) \in \mathcal{A}ct(P_1)$ and $(\alpha, \cdot) \in \mathcal{A}ct(P_2)$.

Recall that for a general GPEPA model $G = M_1 \underset{L}{\bowtie} M_2$, $(H, P) \in \mathcal{B}(G)$ and $\mathsf{N} \in \mathcal{B}(G) \to \mathbb{R}_+$, the component rate function (Definition 3.2.4) is defined as:

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

The term $\frac{\mathcal{R}_\alpha(M_i,\mathsf{N},H,P)}{r_\alpha(M_i,\mathsf{N})}\min(r_\alpha(M_1,\mathsf{N}),r_\alpha(M_2,\mathsf{N}))$ may introduce rational functions of the component counts into the component rate function and thus into the system of differential equations associated with the model. The desirable property of a split-free model, on the other hand, is that the rational expression will always cancel, that is, $\mathcal{R}_\alpha(M_i,\mathsf{N},H,P) = r_\alpha(M_i,\mathsf{N})$ for all $\alpha \in L$. This follows immediately from the definition of a split-free model and that of the component rate function.

For a general split-free GPEPA model G, there may be cooperation between component groups, and we see by Definition 3.2.4 that the general form of the component rate function may include minimum functions and linear combinations (but not rational functions) of component counts, i.e. it is a piecewise linear function of the component counts. Since the only source of non-linearity is the use of minimum functions, we see that Equation 3.5 matches that of Definition 3.3.1 after the following approximation is applied, perhaps more than once in a recursive fashion:

$$\mathbb{E}[\min(r_\alpha(M_1,\mathsf{N}_t),r_\alpha(M_2,\mathsf{N}_t))] \approx \min(\mathbb{E}[r_\alpha(M_1,\mathsf{N}_t)],\mathbb{E}[r_\alpha(M_2,\mathsf{N}_t)])$$

where $M_1 \underset{L}{\bowtie} M_2$ is some submodel of G and $\alpha \in L$. Note in general that in fact:

$$\mathbb{E}[\min(r_\alpha(M_1,\mathsf{N}_t),r_\alpha(M_2,\mathsf{N}_t))] \leq \min(\mathbb{E}[r_\alpha(M_1,\mathsf{N}_t)],\mathbb{E}[r_\alpha(M_2,\mathsf{N}_t)])$$

Any component count configuration which causes the two arguments of a minimum term in the right-hand side of a GPEPA model's associated system of ODEs to be equal is termed a *switch point*. The formal definition follows.

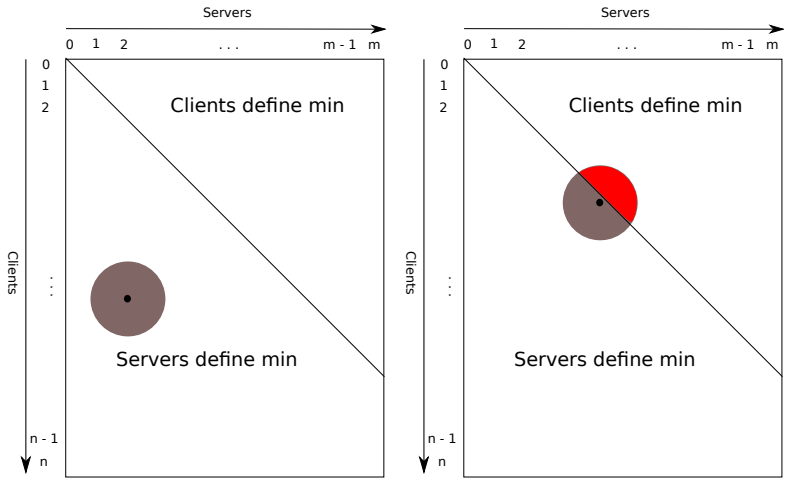
Definition 3.3.4 (GPEPA :: Switch point)**.** *An aggregated state* $\mathsf{N} \in \mathcal{B}(G) \to \mathbb{Z}_+$ *of some GPEPA model* G *is a* switch point *if and only if, for some submodel* $M_1 \underset{L}{\bowtie} M_2$ *of* G, $r_\alpha(M_1,\mathsf{N}) = r_\alpha(M_2,\mathsf{N})$ *for some* $\alpha \in L$.

In the context of the client–server example model $\boldsymbol{CS}(n,m)$, one source of switch points is thus the approximation $\mathbb{E}[\min(N_{\boldsymbol{C}}(t),N_{\boldsymbol{S}}(t))] \approx \min(\mathbb{E}[N_{\boldsymbol{C}}(t)],\mathbb{E}[N_{\boldsymbol{S}}(t)])$.

It is easy to see that far away from switch points, we would expect the ODE approximation to remain good unless the variabilities of the component counting stochastic processes are high. Figure 3.2a illustrates this scenario. The black dot illustrates the expected values of the number of clients and servers with the grey area representing informally the extent of the variability or spread of the distributions. The diagonal line splits the state space into the areas in which each side of the $\min(\cdot,\cdot)$ is the defining term, that is, it is the line of switch points.

Around the switch points, the quality of the approximation decreases with increasing variability of the counting stochastic processes in question. This is illustrated in Figure 3.2b. The red part of the circle illustrates the part of the distribution which has been incorrectly assigned to the servers rather than the clients under the approximation $\mathbb{E}[\min(N_{\boldsymbol{C}}(t),N_{\boldsymbol{S}}(t))] \approx \min(\mathbb{E}[N_{\boldsymbol{C}}(t)],\mathbb{E}[N_{\boldsymbol{S}}(t)])$.

In the next section, we will illustrate the first-moment approximation for the example model $\boldsymbol{CS}(n,m)$ and discuss more practically the usefulness of switch points in predicting the accuracy of the approximation.



(a) Far from a switch point. (b) Near to a switch point.

Figure 3.2: Accuracy of the approximation: $\mathbb{E}[\min(N_{\boldsymbol{C}}(t), N_{\boldsymbol{S}}(t))] \approx \min(\mathbb{E}[N_{\boldsymbol{C}}(t)], \mathbb{E}[N_{\boldsymbol{S}}(t)])$ for the client–server model $\boldsymbol{CS}(n, m)$.

3.3.2 First-moment approximation for running example

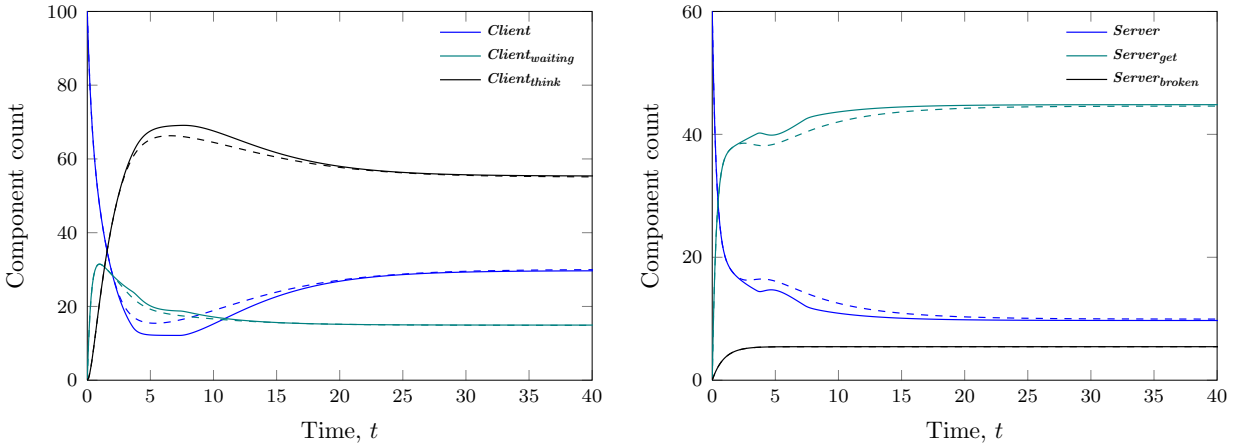
In this section, we will illustrate the first-moment differential equation approximation on the GPEPA model $\boldsymbol{CS}(n, m)$ introduced in Section 3.2.3. Recall that applying Definition 3.3.1 to the model results in the system of differential equations given earlier in Equation 3.4.

Figure 3.3 shows a comparison of the solutions to these ODEs with the actual component count expectations as computed by repeated stochastic simulation. We see that the approximation error is significantly higher in the approximate time interval $[2.5, 15]$. This coincides with close proximity of the component counts to the ***Client*/*Server*** switch point and thus is due to the error in approximating terms $\mathbb{E}[\min(N_{\boldsymbol{C}}(t), N_{\boldsymbol{S}}(t))]$ to construct the differential equations given above. To see this more clearly, Figure 3.4 shows a parametric plot where the error in the approximation of ***Client*** component count expectation is plotted with respect to the distance from the ***Client*/*Server*** switch point, where, for the purposes of this plot, "switch point distance" refers to the quantity $|v_{\boldsymbol{C}}(t) - v_{\boldsymbol{S}}(t)|$.

3.4 Approximation of higher moments

In this section, we will show how the system of ODEs given in Definition 3.3.1 can be extended defining similar differential equations which approximate (joint) higher-order moments of the counts of fluid components. These can be used to provide inexpensive access to key characteristics of the probability distribution of the component counts, such as their variance.

In order to proceed, we will need to characterise how fluid components evolve jointly. To this end, we will generalise the notion of the component rate function for a GPEPA model, a timed action type and *one* given fluid component (Definition 3.2.4) to the *joint component rate function* for a given *set* of fluid components. This will arise naturally in the consideration of higher-order moments involving more than one fluid component count.



(a) Components in **Clients** group

(b) Components in **Servers** group

Figure 3.3: Comparison of component count expectations as computed by ODEs (solid lines) and stochastic simulation (dashed lines) for the model $\boldsymbol{CS}(100, 60)$. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 1.0$, $r_{rst} = 1.0$.

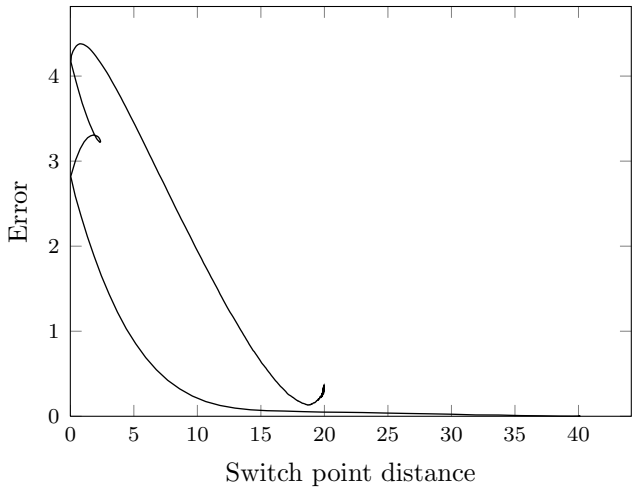


Figure 3.4: Parametric phase plot of ODE expectation approximation (absolute) error against switch point distance for the ***Client*** component count in the model $\boldsymbol{CS}(100, 60)$. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 1.0$, $r_{rst} = 1.0$.

3.4.1 Joint evolution rates of GPEPA models

Informally, the joint component rate is the aggregate rate at which all of the given fluid components complete an action of a given action type in cooperation together. It is the sum of the rates of all transitions of that action type in the aggregated CTMC, in which the specified components are the *only* joint participants. The formal definition of the joint component rate function now follows.

Definition 3.4.1 (GPEPA :: Joint component rate function)**.** *Let G be a GPEPA model. Let $J \subseteq \mathcal{B}(G)$ be a non-empty set of fluid component derivative states. Let $\alpha \in \mathcal{A}_t$ be a timed action type and let $\mathsf{N} \in \mathcal{B}(G) \to \mathbb{R}_+$. Then the joint component rate is $\mathcal{R}_\alpha(G, \mathsf{N}, J)$, defined as follows.*

$$\mathcal{R}_\alpha(M_1 \underset{L}{\bowtie} M_2, \mathsf{N}, J) := \begin{cases} \frac{\mathcal{R}_\alpha(M_1,\mathsf{N},J_1)}{r_\alpha(M_1,\mathsf{N})} \frac{\mathcal{R}_\alpha(M_2,\mathsf{N},J_2)}{r_\alpha(M_2,\mathsf{N})} \min(r_\alpha(M_1,\mathsf{N}), r_\alpha(M_2,\mathsf{N})) & \text{if } \alpha \in L,\ J \nsubseteq \mathcal{B}(M_j) \\ & \quad \text{for } j = 1 \text{ and } 2 \\ \mathcal{R}_\alpha(M_j, \mathsf{N}, J) & \text{if } \alpha \notin L,\ J \subseteq \mathcal{B}(M_j) \\ & \quad \text{for } j = 1 \text{ or } 2 \\ 0 & \text{otherwise} \end{cases}$$

In the first line of the definition, we define J_1 and J_2 to be the unique partition of J such that $J_1 \subseteq \mathcal{B}(M_1)$ and $J_2 \subseteq \mathcal{B}(M_2)$. As before, terms with zeros in the denominator are defined as zero. Finally:

$$\mathcal{R}_\alpha(H\{D\}, \mathsf{N}, J) := \begin{cases} \mathsf{N}(H,P)\, r_\alpha(P) & \text{if } J = \{(H,P)\} \\ 0 & \text{otherwise} \end{cases}$$

In the final case of the definition, the cardinality of J has to be 1 to return a non-zero value, since there is no cooperation between the components of a component group by definition, and we are computing the *joint* rate of evolution between cooperating components.

We now wish to relate the joint component rate function to the underlying CTMC in a similar fashion to Theorem 3.2.7 for the component rate function. In order to do this, we define the *joint evolution set* $\mathcal{J}(G)$ of a GPEPA model G. This set enumerates the possible ways in which fluid components can evolve together with a particular action type. It consists of 3-tuples $(J_-, J_+, \alpha) \in \mathcal{P}(\mathcal{B}(G)) \times \mathcal{P}(\mathcal{B}(G)) \times \mathcal{A}_t$. Each such tuple represents an α-transition which evolves the fluid components in J_- into those in J_+. The formal definition follows.

Definition 3.4.2 (GPEPA :: Joint evolution set)**.** *Let G be a GPEPA model. Then the joint evolution set is $\mathcal{J}(G)$, defined as follows.*

$$\begin{aligned} \mathcal{J}(M_1 \underset{L}{\bowtie} M_2) := {} & \{(J_-^1 \cup J_-^2, J_+^1 \cup J_+^2, \alpha) \,:\, (J_-^1, J_+^1, \alpha) \in \mathcal{J}(M_1),\ (J_-^2, J_+^2, \alpha) \in \mathcal{J}(M_2),\ \alpha \in L\} \\ & \cup \{(J_-, J_+, \alpha) \,:\, (J_-, J_+, \alpha) \in \mathcal{J}(M_1)\ ,\ \alpha \notin L\} \\ & \cup \{(J_-, J_+, \alpha) \,:\, (J_-, J_+, \alpha) \in \mathcal{J}(M_2)\ ,\ \alpha \notin L\} \\ \mathcal{J}(H\{D\}) := {} & \bigcup_{\substack{P \xrightarrow{(\alpha,\cdot)} Q \\ P,Q \in \mathcal{B}(G,H)}} \{\{(H,P)\}, \{(H,Q)\}, \alpha\} \end{aligned}$$

The following theorem then gives the rates associated with transitions specified by the elements of $\mathcal{J}(G)$.

Theorem 3.4.3. *Let G be a GPEPA model and let $\mathsf{N}, \mathsf{N}' \in \mathcal{B}(G) \to \mathbb{Z}_+$ represent states of the aggregated CTMC of G. Let $\alpha \in \mathcal{A}_t$ be a timed action type.*

Then the sum of the rates of all α-transitions from N to N' is the sum of all quantities:

$$\rho_\alpha(J_-, J_+)\mathcal{R}_\alpha(G, \mathsf{N}, J_-) \tag{3.7}$$

taken over all elements $(J_-, J_+, \alpha) \in \mathcal{J}(G)$, which characterise N', that is, all elements for which:

1. $\mathsf{N}'(H,P) = \mathsf{N}(H,P) - 1$ *for all* $(H,P) \in J_- \setminus J_+$,
2. $\mathsf{N}'(H,P) = \mathsf{N}(H,P) + 1$ *for all* $(H,P) \in J_+ \setminus J_-$ *and*
3. $\mathsf{N}'(H,P) = \mathsf{N}(H,P)$ *for all* $(H,P) \in \mathcal{B}(G) \setminus (J_- \ominus J_+)$.[6]

Each term given by Equation 3.7 is zero if no such α-transitions exist.

Proof. See Hayden and Bradley [139, Theorem 5.3]. □

The expression in Equation 3.7 uses an extended version of the derivative weighting function originally given by Definition 3.2.5. Its purpose is to compute the probability that after a joint α-action, a given set of fluid components transit together to another given set of fluid components. We give the extension in the following definition.

Definition 3.4.4 (GPEPA :: Joint derivative weighting function)**.** *Let G be a GPEPA model and let $(J_-, J_+, \alpha) \in \mathcal{J}(G)$. Then $\rho_\alpha(J_-, J_+)$ is the joint derivative weighting function, defined as follows.*

$$\rho_\alpha(J_-, J_+) := \prod_{(H,P) \in J_-} p_\alpha(P, Q_{H,J_+})$$

where Q_{H,J_+} is defined as the unique fluid component Q, such that $(H,Q) \in J_+$. That it will exist and be unique is guaranteed by the fact that $(J_-, J_+, \alpha) \in \mathcal{J}(G)$.

3.4.2 Action of the infinitesimal generator on higher-order moments

Making use of the characterisation of a GPEPA model's underlying CTMC given by Theorem 3.4.3, we intend to construct differential equations approximating any joint moment of fluid component counts. We will achieve this by generalising the approach of Section 3.3.1. Specifically, we are now interested in the action of the infinitesimal generator on products of component counts.

[6] Where $A \ominus B := (A \cup B) \setminus (A \cap B)$ is the symmetric difference of the two sets A and B.

Let G be a GPEPA model and $\mathsf{M} \in \mathcal{B}(G) \to \mathbb{Z}_+$ specify a particular joint moment of fluid component counting stochastic processes that is of interest $\mathbb{E}[M(t)]$ by $M(t) := \prod_{B \in \mathcal{B}(G)} N_B(t)^{\mathsf{M}(B)}$. Using Theorem 3.4.3, we may then see that the action of the infinitesimal generator $\mathbf{A}_G$ on $M(t)$ is given by:

$$\mathbf{A}_G\, M(t) = \sum_{(J_-,J_+,\alpha)\in\mathcal{J}(G)} \rho_\alpha(J_-,J_+)(F_+(t)F_-(t)F(t) - M(t))\mathcal{R}_\alpha(G,\mathsf{N}_t,J_-)$$

where:

$$F_\pm(t) := \prod_{B\in J_\pm\setminus J_\mp} (N_B(t)\pm 1)^{\mathsf{M}(B)} \qquad \text{and} \qquad F(t) := \prod_{B\in\mathcal{B}(G)\setminus(J_-\ominus J_+)} N_B(t)^{\mathsf{M}(B)}$$

and for $t \in \mathbb{R}_+$, $\mathsf{N}_t \in \mathcal{B}(G) \to \mathbb{Z}_+$ is given by $\mathsf{N}_t(B) := N_B(t)$ for all $B \in \mathcal{B}(G)$. Note that we have suppressed the dependence of $F_+(t)$, $F_-(t)$ and $F(t)$ on J_- and J_+ simply to ease the notation. By way of example, consider the joint moment $\mathbb{E}[N_{\boldsymbol{Cw}}(t)N_{\boldsymbol{Sg}}(t)]$ in the context of the client–server model $\boldsymbol{CS}(n,m)$. Then if $J_- = \{(\textbf{Clients}, \boldsymbol{Client}), (\textbf{Servers}, \boldsymbol{Server})\}$ and $J_+ = \{(\textbf{Clients}, \boldsymbol{Client_{waiting}}), (\textbf{Servers}, \boldsymbol{Server_{get}})\}$, we have $F_+(t) = (N_{\boldsymbol{Cw}}(t)+1)(N_{\boldsymbol{Sg}}(t)+1)$ and $F_-(t) = F(t) = 1$.

Applying Equation 2.1 then yields the exact differential equation:

$$\dot{\mathbb{E}}[M(t)] = \sum_{(J_-,J_+,\alpha)\in\mathcal{J}(G)} \rho_\alpha(J_-,J_+)\mathbb{E}\left[(F_+(t)F_-(t)F(t) - M(t))\mathcal{R}_\alpha(G,\mathsf{N}_t,J_-)\right] \tag{3.8}$$

In fact, this equation can also be derived using an informal intuitive argument. Each transition in the aggregated CTMC corresponds to one particular pair $(J_-,J_+,\alpha) \in \mathcal{J}(G)$, depending on the changes it makes to fluid component counts; and a particular timed action type $\alpha \in \mathcal{A}_t$. As we know from Theorem 3.4.3, the sum of the rates of all such transitions is the quantity $\rho_\alpha(J_-,J_+)\mathcal{R}_\alpha(G,\mathsf{N}_t,J_-)$. The term $F_+(t)F_-(t)F(t) - M(t)$ is simply *the change in $M(t)$ due to such a transition*, so the above equation can be derived informally by considering the *change in $M(t)$* multiplied by the *rate at which it occurs*, summed over all enabled transitions.

As it stands, the differential equation in Equation 3.8 is not something we can solve since to do so would require a differential equation for the term $\mathbb{E}[(F_+(t)F_-(t)F(t) - M(t))\mathcal{R}_\alpha(G,\mathsf{N}_t,J_-)]$ which appears in its right-hand side. A more practical approach is to rewrite this term and apply approximations where necessary so that it can be written as a function of component-count moments. We show how this approach leads to a system of ODEs which we can solve in the next section.

3.4.3 Derivation of the approximating ODEs

The previous section presented an equation for the derivative of a general moment of component-counting processes (Equation 3.8). In this section, we show how an approximation to this equation can be constructed using a suitable system of ODEs. In order to achieve this, we must rearrange Equation 3.8 to make the individual component-count moments explicit on the right-hand side. Each of these component-count moments is then approximated by its own differential equation (as finally achieved in Equation 3.13).

We begin by using the binomial theorem to see that the term $F_+(t)F_-(t)F(t)$ of Equation 3.8 is the sum of all monomials of the form:

$$\prod_{B\in J_-\setminus J_+}(-1)^{\mathsf{M}(B)-\mathsf{K}(B)}\prod_{B\in\mathcal{B}(G)}\binom{\mathsf{M}(B)}{\mathsf{K}(B)}N_B(t)^{\mathsf{K}(B)} \tag{3.9}$$

taken over all elements $\mathsf{K}\in\mathcal{B}(G)\to\mathbb{Z}_+$, which satisfy $0\leq\mathsf{K}(B)\leq\mathsf{M}(B)$ for all $B\in J_-\ominus J_+$, and $\mathsf{K}(B)=\mathsf{M}(B)$ for all $B\in\mathcal{B}(G)\setminus(J_-\ominus J_+)$.

Now consider the term $F_+(t)F_-(t)F(t)-M(t)$ and note that $M(t)$ simply serves to cancel the highest-order such monomial, i.e. the case of $\mathsf{K}(B)=\mathsf{M}(B)$ for all $B\in\mathcal{B}(G)$. Therefore, after this cancellation, all monomials in the sum will have order at most $M-1$, where $M:=\sum_{B\in\mathcal{B}(G)}\mathsf{M}(B)$ is the order of the moment of interest. That is, $F_+(t)F_-(t)F(t)-M(t)$ is equal to the sum of all monomials of the form of Equation 3.9 taken over all elements $\mathsf{K}\in\mathcal{B}(G)\to\mathbb{Z}_+$, which satisfy $0\leq\mathsf{K}(B)\leq\mathsf{M}(B)$ for all $B\in J_-\ominus J_+$; and $\mathsf{K}(B)=\mathsf{M}(B)$ for all $B\in\mathcal{B}(G)\setminus(J_-\ominus J_+)$ where $\sum_{B\in\mathcal{B}(G)}\mathsf{K}(B)\neq M$. We denote this subset of $\mathcal{B}(G)\to\mathbb{Z}_+$ by $\mathcal{K}(J_-,J_+,\mathsf{M})$.

Consider now the expectation term:

$$\mathbb{E}\left[(F_+(t)F_-(t)F(t)-M(t))\mathcal{R}_\alpha(G,\mathsf{N}_t,J_-)\right] \tag{3.10}$$

from Equation 3.8. Using the fact above that $F_+(t)F_-(t)F(t)-M(t)$ consists of a sum of monomials of the form of Equation 3.9, we may expand Equation 3.10 into a sum of expectations, one for each monomial term, i.e. it is equal to the sum of the terms:

$$\left(\prod_{B\in J_-\setminus J_+}(-1)^{\mathsf{M}(B)-\mathsf{K}(B)}\prod_{B\in\mathcal{B}(G)}\binom{\mathsf{M}(B)}{\mathsf{K}(B)}\right)\mathbb{E}\left[\prod_{B\in\mathcal{B}(G)}N_B(t)^{\mathsf{K}(B)}\mathcal{R}_\alpha(G,\mathsf{N}_t,J_-)\right] \tag{3.11}$$

taken over all $\mathsf{K}\in\mathcal{K}(J_-,J_+,\mathcal{M})$.

Before we proceed, we note that the joint component rate function enjoys the same homogeneity property as was shown for the component rate function in Lemma 3.3.2. That is, for any $\mathsf{N}\in\mathcal{B}(G)\to\mathbb{R}_+$ and $C\in\mathbb{R}_+$:

$$C\times\mathcal{R}_\alpha(G,\mathsf{N},J)=\mathcal{R}_\alpha(G,\mathsf{N}^*,J)$$

where $\mathsf{N}^*\in\mathcal{B}(G)\to\mathbb{R}_+$ is defined by $\mathsf{N}^*(B):=C\times\mathsf{N}(B)$ for all $B\in\mathcal{B}(G)$. This allows us to write Equation 3.11 as:

$$\left(\prod_{B\in J_-\setminus J_+}(-1)^{\mathsf{M}(B)-\mathsf{K}(B)}\prod_{B\in\mathcal{B}(G)}\binom{\mathsf{M}(B)}{\mathsf{K}(B)}\right)\mathbb{E}\left[\mathcal{R}_\alpha(G,\mathsf{N}_t^*[\mathsf{K}],J_-)\right] \tag{3.12}$$

where for $t\in\mathbb{R}_+$, $\mathsf{N}_t^*[\mathsf{K}]\in\mathcal{B}(G)\to\mathbb{Z}_+$ is given by:

$$\mathsf{N}_t^*[\mathsf{K}](B):=N_B(t)\times\prod_{B'\in\mathcal{B}(G)}N_{B'}(t)^{\mathsf{K}(B')}$$

for all $B\in\mathcal{B}(G)$.

Now for a given moment, specified by $\mathsf{M} \in \mathcal{B}(G) \to \mathbb{Z}_+$, write $v_{\mathsf{M}}(t)$ as its ODE approximation, to be defined shortly. For split-free models, the joint component rate function (Definition 3.4.1), as in the case of the component rate function (Definition 3.2.4), may involve minimum functions and linear combinations (but not rational functions) of component counts. So applying the approximation $\mathbb{E}[\min(\cdot,\cdot)] \approx \min(\mathbb{E}[\cdot], \mathbb{E}[\cdot])$, potentially repeatedly to Equation 3.12, yields in terms of the ODE approximations:

$$\left(\prod_{B \in J_- \setminus J_+} (-1)^{\mathsf{M}(B)-\mathsf{K}(B)} \prod_{B \in \mathcal{B}(G)} \binom{\mathsf{M}(B)}{\mathsf{K}(B)} \right) \mathcal{R}_\alpha(G, \mathsf{V}_t^*[\mathsf{K}], J_-)$$

where for $t \in \mathbb{R}_+$, $\mathsf{V}_t^*[\mathsf{K}] \in \mathcal{B}(G) \to \mathbb{R}_+$ is given by, for all $B \in \mathcal{B}(G)$:

$$\mathsf{V}_t^*[\mathsf{K}](B) := v_{\mathsf{I}}(t)$$

where $\mathsf{I}(B') = \mathsf{K}(B')$ for all $B' \neq B$ and $\mathsf{I}(B) = \mathsf{K}(B) + 1$. Note that since $\sum_{B' \in \mathcal{B}(G)} \mathsf{K}(B') \leq M - 1$ for all $\mathsf{K} \in \mathcal{K}(J_-, J_+, \mathsf{M})$, ODE approximations of joint moments of order at most M occur in such terms.

Combining these terms to approximate Equation 3.8 gives the following ODE where the right-hand side is in terms of the ODE approximations of the joint moments of order M and below.

Definition 3.4.5 (GPEPA :: Higher-order moment ODEs)**.** *Let G be a GPEPA model. We define the evolution of the $v_{\mathsf{M}}(t)$ over time for $\mathsf{M} \in \mathcal{B}(G) \to \mathbb{Z}_+$ by the system of first-order coupled ODEs:*

$$\begin{aligned} \dot{v}_{\mathsf{M}}(t) = & \sum_{(J_-,J_+,\alpha) \in \mathcal{J}(G)} \rho_\alpha(J_-, J_+) \\ & \sum_{\mathsf{K} \in \mathcal{K}(J_-,J_+,\mathsf{M})} \left(\left(\prod_{B \in J_- \setminus J_+} (-1)^{\mathsf{M}(B)-\mathsf{K}(B)} \prod_{B \in \mathcal{B}(G)} \binom{\mathsf{M}(B)}{\mathsf{K}(B)} \right) \mathcal{R}_\alpha(G, \mathsf{V}_t^*[\mathsf{K}], J_-) \right) \end{aligned} \tag{3.13}$$

The initial condition for each moment, $\mathsf{M} \in \mathcal{B}(G) \to \mathbb{Z}_+$, is that naturally implied by the initial state of G. Specifically, we define:

$$v_{\mathsf{M}}(0) := \prod_{B \in \mathcal{B}(G)} \mathcal{C}(G, B)^{\mathsf{M}(B)}$$

We note here that in the case of first-order moments, Equation 3.13 reduces to the ODE given by Definition 3.3.1. Also for models which are not split-free, all of the above still holds but the nature of the approximation is more complicated, as in the first-order case.

3.4.4 Standard-deviation approximation for running example

In this section, we will illustrate the higher-moment differential-equation approximations on the GEPA model $\boldsymbol{CS}(n, m)$ introduced in Section 3.2.3. Specifically, we will show how we may use the second-order moment approximations to approximate the variance and thus the standard deviation of fluid component counts in the model. We will write $v_{\boldsymbol{C}^2}(t)$ for the ODE approximation to the second moment of the ***Client***-counting process $\mathbb{E}[N_{\boldsymbol{C}}^2(t)]$ and $v_{\boldsymbol{C}\cdot\boldsymbol{Sg}}(t)$

for the ODE approximation to the joint moment of the $\boldsymbol{Client}$- and $\boldsymbol{Server_{get}}$-counting processes $\mathbb{E}[N_{\boldsymbol{C}}(t)N_{\boldsymbol{Sg}}(t)]$, and we adopt a similar notation for the other second-order moment approximations.

In order to proceed, we wish to evaluate Equation 3.13 for each second-order moment. We begin by enumerating the joint evolution set and the corresponding joint component rates:[7]

1. *request*-transitions of $J_- = \{\boldsymbol{Client}, \boldsymbol{Server}\}$ into $J_+ = \{\boldsymbol{Client_{waiting}}, \boldsymbol{Server_{get}}\}$ occur at rate:
$$\rho_\alpha(J_-, J_+)\mathcal{R}_{request}(\boldsymbol{CS}(n,m), \mathsf{N}_t, J_-) = \min(N_{\boldsymbol{C}}(t), N_{\boldsymbol{S}}(t))r_r$$
2. *data*-transitions of $J_- = \{\boldsymbol{Client_{waiting}}, \boldsymbol{Server_{get}}\}$ into $J_+ = \{\boldsymbol{Client_{think}}, \boldsymbol{Server}\}$ occur at rate:
$$\rho_\alpha(J_-, J_+)\mathcal{R}_{data}(\boldsymbol{CS}(n,m), \mathsf{N}_t, J_-) = \min(N_{\boldsymbol{Cw}}(t), N_{\boldsymbol{Sg}}(t))r_d$$
3. *timeout*-transitions of $J_- = \{\boldsymbol{Client_{waiting}}\}$ into $J_+ = \{\boldsymbol{Client}\}$ occur at rate:
$$\rho_\alpha(J_-, J_+)\mathcal{R}_{timeout}(\boldsymbol{CS}(n,m), \mathsf{N}_t, J_-) = N_{\boldsymbol{Cw}}(t)r_{tmt}$$
4. *think*-transitions of $J_- = \{\boldsymbol{Client_{think}}\}$ into $J_+ = \{\boldsymbol{Client}\}$ occur at rate:
$$\rho_\alpha(J_-, J_+)\mathcal{R}_{think}(\boldsymbol{CS}(n,m), \mathsf{N}_t, J_-) = N_{\boldsymbol{Ct}}(t)r_t$$
5. *break*-transitions of $J_- = \{\boldsymbol{Server}\}$ into $J_+ = \{\boldsymbol{Server_{broken}}\}$ occur at rate:
$$\rho_\alpha(J_-, J_+)\mathcal{R}_{break}(\boldsymbol{CS}(n,m), \mathsf{N}_t, J_-) = N_{\boldsymbol{S}}(t)r_b$$
6. *break*-transitions of $J_- = \{\boldsymbol{Server_{get}}\}$ into $J_+ = \{\boldsymbol{Server_{broken}}\}$ occur at rate:
$$\rho_\alpha(J_-, J_+)\mathcal{R}_{break}(\boldsymbol{CS}(n,m), \mathsf{N}_t, J_-) = N_{\boldsymbol{Sg}}(t)r_b$$
7. *reset*-transitions of $J_- = \{\boldsymbol{Server_{broken}}\}$ into $J_+ = \{\boldsymbol{Server}\}$ occur at rate:
$$\rho_\alpha(J_-, J_+)\mathcal{R}_{reset}(\boldsymbol{CS}(n,m), \mathsf{N}_t, J_-) = N_{\boldsymbol{Sb}}(t)r_{rst}$$

Let us consider the joint second-order moment $\mathbb{E}[N_{\boldsymbol{Cw}}(t)N_{\boldsymbol{Sg}}(t)]$. The only transitions of interest are 1, 2, 3 and 6 in the above enumeration since they are the only ones for which either:

$$\boldsymbol{Client_{waiting}} \in J_- \ominus J_+ \text{ or } \boldsymbol{Server_{get}} \in J_- \ominus J_+$$

Indeed, if $J_- \ominus J_+$ does not include at least one $B \in \mathcal{B}(\boldsymbol{CS}(n,m))$ for which $\mathsf{M}(B) > 0$, $\mathcal{K}(J_-, J_+, \mathsf{M}) = \emptyset$ and the pair (J_-, J_+) would contribute nothing to the right-hand side of Equation 3.13.

Now for transition 1, $\mathcal{K}(J_-, J_+, \mathsf{M})$ consists of three elements, K_1, K_2 and K_3, where $\mathsf{K}_1(B) = 0$ for all $B \in \mathcal{B}(\boldsymbol{CS}(n,m))$, $\mathsf{K}_2(\boldsymbol{Client_{waiting}}) = 1$, $\mathsf{K}_2(\boldsymbol{Server_{get}}) = 0$ and $\mathsf{K}_2(B) = 0$ otherwise, and $\mathsf{K}_3(\boldsymbol{Client_{waiting}}) = 0$, $\mathsf{K}_3(\boldsymbol{Server_{get}}) = 1$ and $\mathsf{K}_3(B) = 0$ otherwise. Then K_1

[7] For brevity, we drop the component group labels when giving the J_- and J_+ sets.

contributes $\min(v_{\boldsymbol{C}}(t), v_{\boldsymbol{S}}(t))r_r$ to Equation 3.13, K_2 contributes $\min(v_{\boldsymbol{Cw}\cdot\boldsymbol{C}}(t), v_{\boldsymbol{Cw}\cdot\boldsymbol{S}}(t))r_r$ and K_3 contributes: $\min(v_{\boldsymbol{C}\cdot\boldsymbol{Sg}}(t), v_{\boldsymbol{S}\cdot\boldsymbol{Sg}}(t))r_r$. So together, transition 1 contributes to Equation 3.13:

$$(\min(v_{\boldsymbol{C}}(t), v_{\boldsymbol{S}}(t)) + \min(v_{\boldsymbol{C}\cdot\boldsymbol{Cw}}(t), v_{\boldsymbol{Cw}\cdot\boldsymbol{S}}(t)) + \min(v_{\boldsymbol{C}\cdot\boldsymbol{Sg}}(t), v_{\boldsymbol{S}\cdot\boldsymbol{Sg}}(t)))r_r$$

For transition 2, $\mathcal{K}(J_-, J_+, \mathsf{M})$ is as for transition 1 and the contribution to Equation 3.13 is similarly:

$$(\min(v_{\boldsymbol{Cw}}(t), v_{\boldsymbol{Sg}}(t)) - \min(v_{\boldsymbol{Cw}^2}(t), v_{\boldsymbol{Cw}\cdot\boldsymbol{Sg}}(t)) - \min(v_{\boldsymbol{Cw}\cdot\boldsymbol{Sg}}(t), v_{\boldsymbol{Sg}^2}(t)))r_d$$

For transition 3, $\mathcal{K}(J_-, J_+, \mathsf{M})$ consists of just one element, K_1, where $\mathsf{K}_1(\boldsymbol{Server}_{\boldsymbol{get}}) = 1$ and $\mathsf{K}_1(B) = 0$ otherwise. The contribution to Equation 3.13 is then:

$$-v_{\boldsymbol{Cw}\cdot\boldsymbol{Sg}}(t)r_{tmt}$$

For transition 6, $\mathcal{K}(J_-, J_+, \mathsf{M})$ consists of just one element, K_1, where $\mathsf{K}_1(\boldsymbol{Client}_{\boldsymbol{waiting}}) = 1$ and $\mathsf{K}_1(B) = 0$ otherwise. The contribution to Equation 3.13 is then:

$$-v_{\boldsymbol{Cw}\cdot\boldsymbol{Sg}}(t)r_b$$

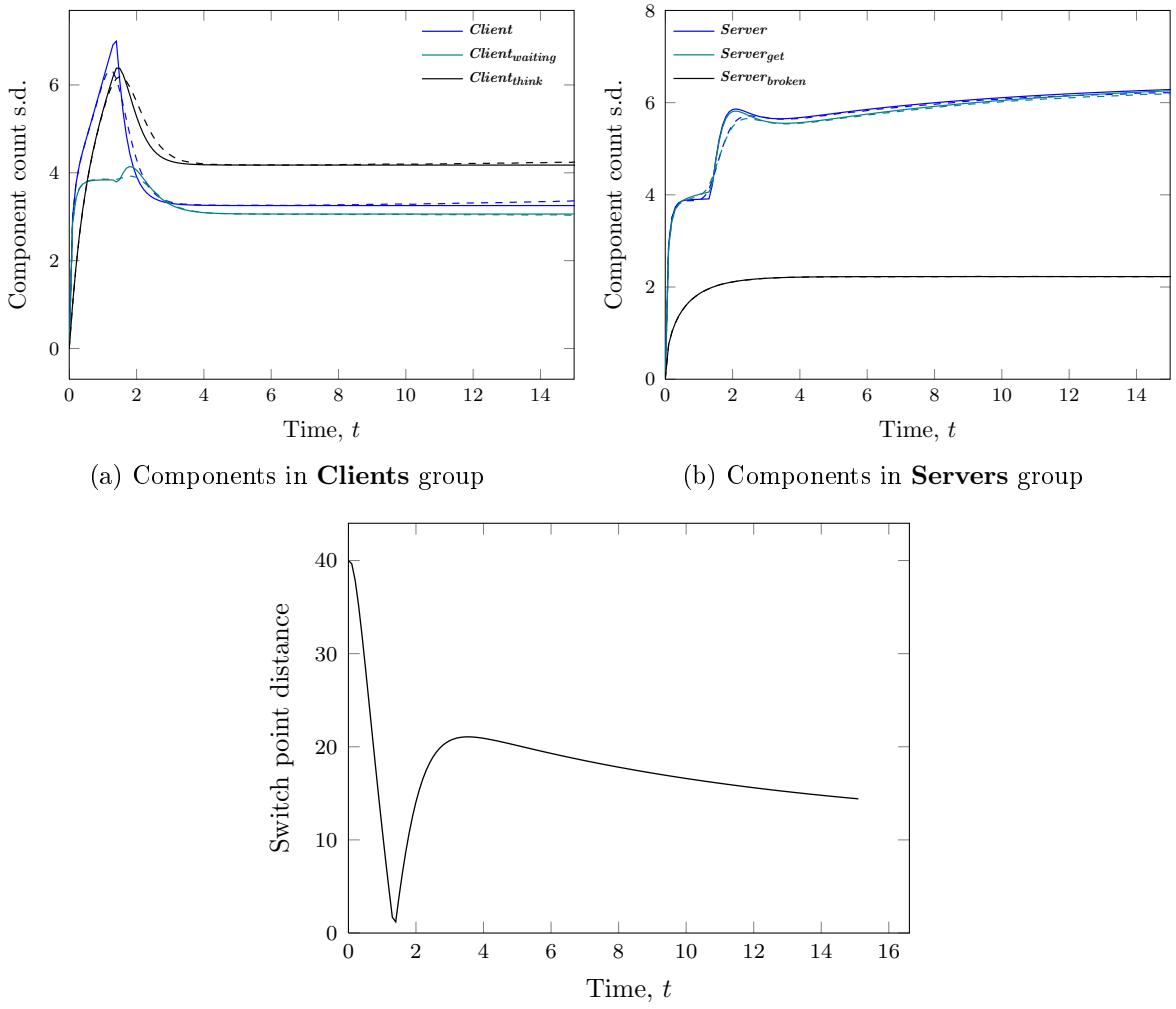
Collecting all of these individual contributions to form the ODE of Equation 3.13 for $\dot{v}_{\boldsymbol{Cw}\cdot\boldsymbol{Sg}}(t)$ gives:

$$\begin{aligned}\dot{v}_{\boldsymbol{Cw}\cdot\boldsymbol{Sg}}(t) = {} & (\min(v_{\boldsymbol{C}}(t), v_{\boldsymbol{S}}(t)) + \min(v_{\boldsymbol{C}\cdot\boldsymbol{Cw}}(t), v_{\boldsymbol{Cw}\cdot\boldsymbol{S}}(t)) + \min(v_{\boldsymbol{C}\cdot\boldsymbol{Sg}}(t), v_{\boldsymbol{S}\cdot\boldsymbol{Sg}}(t)))r_r \\ & + (\min(v_{\boldsymbol{Cw}}(t), v_{\boldsymbol{Sg}}(t)) - \min(v_{\boldsymbol{Cw}^2}(t), v_{\boldsymbol{Cw}\cdot\boldsymbol{Sg}}(t)) - \min(v_{\boldsymbol{Cw}\cdot\boldsymbol{Sg}}(t), v_{\boldsymbol{Sg}^2}(t)))r_d \\ & - v_{\boldsymbol{Cw}\cdot\boldsymbol{Sg}}(t)r_{tmt} - v_{\boldsymbol{Cw}\cdot\boldsymbol{Sg}}(t)r_b \end{aligned} \tag{3.14}$$

Repeating this procedure for the first-order and other joint second-order moments yields a complete system of 27 ODEs, which uniquely determine all 27 approximations. The full system is given in Appendix A.2.1 and is also available online [131].

Figures 3.5a and 3.5b show a comparison of the standard deviation approximations computed by solving these ODEs compared with the actual quantities as computed by repeated stochastic simulation. We see that the approximation error is significantly higher in the approximate time interval $[1.0, 3.5]$ and also towards the end of the depicted range of time. This is particulary evident in Figure 3.5a. As can be seen from Figure 3.5c, this coincides with close proximity of the component counts to the $\boldsymbol{Client}/\boldsymbol{Server}$ switch point and thus is likely due to the error in approximating terms $\mathbb{E}[\min(N_{\boldsymbol{C}}(t), N_{\boldsymbol{S}}(t))]$ to construct the differential equations.

Figure 3.6 depicts similar results for skewness and Figure 3.7 for kurtosis. As for the second-order case, the approximating ODEs are derived directly from Equation 3.13. The number of ODEs required for the third-order analysis is 83, compared with 209 for the fourth-order analysis. For brevity we do not give in this thesis the systems of ODEs which were used to compute these results, but they are available online [131]. The error in these two plots is noticeably higher around the switch point than for the case of the standard deviation or first-order means. However, the approximations still perform generally very well when at a reasonable distance from the switch point. The fact that the error generally gets worse for higher-moment analyses has also been noted in Stefanek et al. [229].

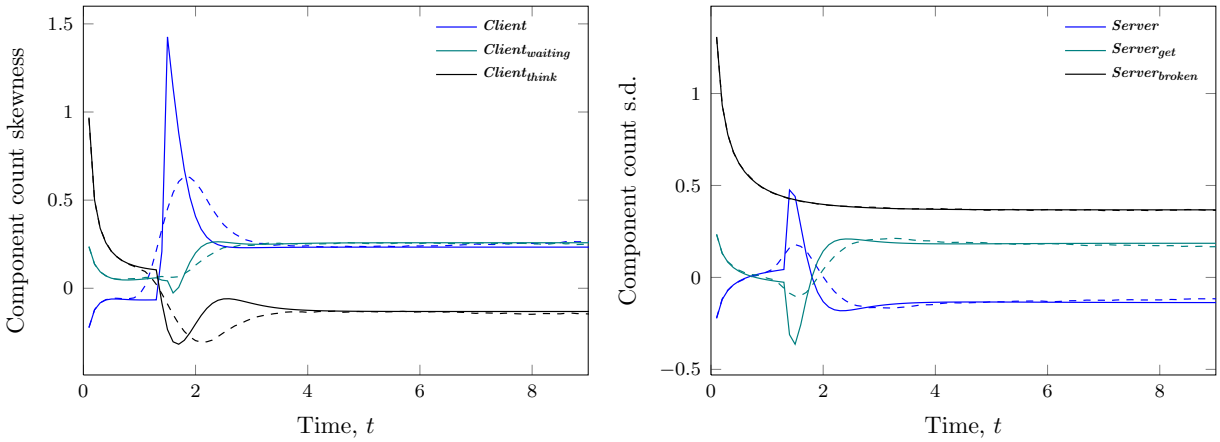


(a) Components in **Clients** group

(b) Components in **Servers** group

(c) ***Client***/***Server*** switch point distance

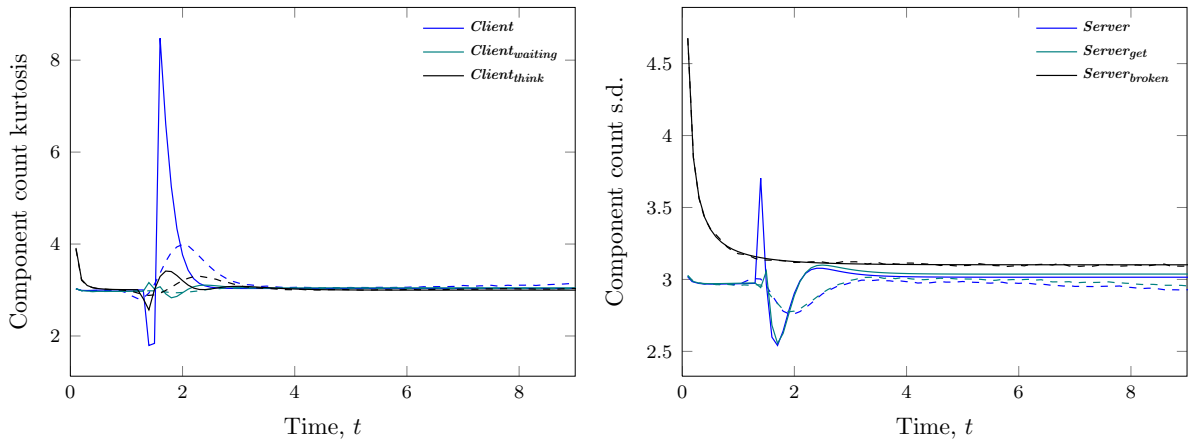
Figure 3.5: Comparison of component-count standard deviations as computed by ODEs (solid lines) and stochastic simulation (dashed lines) for the model $\boldsymbol{CS}(100, 60)$. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 2.0$, $r_{rst} = 1.0$.



(a) Components in **Clients** group

(b) Components in **Servers** group

Figure 3.6: Comparison of component-count skewness computed by ODEs (solid lines) and stochastic simulation (dashed lines) for the model $\boldsymbol{CS}(100, 60)$. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 2.0$, $r_{rst} = 1.0$.



(a) Components in **Clients** group

(b) Components in **Servers** group

Figure 3.7: Comparison of component-count kurtosis computed by ODEs (solid lines) and stochastic simulation (dashed lines) for the model $\boldsymbol{CS}(100, 60)$. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 2.0$, $r_{rst} = 1.0$.

3.5 Tractability of moment approximations

If we are interested in computing Nth-order moment approximations for a GPEPA model G, we must construct the system of ODEs consisting of Equation 3.13 taken for every Mth ($1 \leq M \leq N$) joint moment of component counts. The number of required ODEs in total is then:

$$\sum_{i=1}^{N} \binom{\mathcal{N}(G) + N - 1}{\mathcal{N}(G) - 1} = \binom{\mathcal{N}(G) + n}{\mathcal{N}(G)} - 1 \approx \mathcal{N}(G)^N \text{ when } \mathcal{N}(G) \gg N$$

where we recall that $\mathcal{N}(G) = |\mathcal{B}(G)|$ is the total number of fluid component derivative states in the model. For a fixed order, this effectively polynomial growth in the number of fluid components will obviously have a practical implication for the degree of moment analysis that will be possible for a given model. However first-order, second-order and very often, third-order moment analysis should certainly be very tractable for realistically-detailed models. For example, in the case of the model $\boldsymbol{CS}(n, m)$, $\mathcal{N}(\boldsymbol{CS}(n, m)) = 6$ and to access second-order moments, the total number of ODEs we have to solve is only 27. For third-order moments, this increases to 83 and in the case of fourth-order moments, to 209. Comparing this with the fact that the underlying aggregated state space has of the order of six million states for the fairly modest configuration $n = 100$ and $m = 50$, we see that the approximate moment analysis described here, even up to fourth-order moments and higher is massively more tractable than attempting to exactly solve the CTMC directly using one of the techniques summarised in Section 2.2. Indeed, the key reason for this is that the size of the system of ODEs (for any N) is totally independent of the model's population size, which in this case is $n + m$.

A more detailed model with 25 fluid component derivative states would require 350 ODEs to be solved in order to approximate the second-order moments. For third-order moments, this increases quickly to around $3,000$. However, the underlying aggregated CTMC of such a model has around 10^{25} states (assuming a population of about 100). Third-order (and higher) moment analysis is thus still massively more tractable than an exact approach. For models with 50 fluid component derivative states, second-order moment analysis is still very much a feasible option through solving the system of $1,325$ ODEs. Indeed, the Grouped PEPA Analyser tool [229, 235] which implements the techniques of this chapter is able to solve systems of over $10,000$ moment ODEs in a matter of minutes on a standard desktop computer.

Further examples of third- and fourth-order moment analysis using these techniques and the grouped PEPA analyser tool (GPA) can be found in Stefanek et al. [229]. The use of higher-moments for the analysis of passage-time cumulative distribution functions is discussed in Hayden et al. [142] with further examples in Hayden et al. [141].

3.6 Conclusion

In this chapter, we have introduced grouped PEPA as a simple syntactic extension of the stochastic process algebra PEPA. This provides a framework in which to define ODE-based approximations to general performance models by making an unambiguous distinction between so-called *fluid components* and the rest of the model. This allows us to tackle GPEPA models with fluid components that have internal composition without the need to expand the parallelism explicitly as in Clark et al. [81]. The GPEPA framework also extends the applicability

of ODE-based approximation techniques to a much larger class of models than originally considered by Hillston [147].

Most importantly, we have developed systems of differential equations which can be used to approximate higher-order moments. This can provide inexpensive access to approximations of key stochastic quantities of interest such as the variance of a component count. The variance in turn tells a modeller whether the mean is an accurate prediction of system behaviour. Furthermore, higher moments beyond the variance, can be used to calculate passage-time measures directly. Being able to compute all of these measures by solving a tractable system of ODEs, opens the door to rapid scalability analysis and optimisation over many dimensions of model parameters. These developments have clear application to the analysis of performance models specified in formalisms other than PEPA and could also potentially be used to extend related techniques such as mean-field analysis.

Deriving the moment ODEs from the underlying CTMC in this way, leads to the characterisation of so-called *switch points*. These are a powerful qualitative predictor of the accuracy of the ODE approximation, both in the case of first-order and higher-order moments.

The techniques described in this chapter have been implemented in a freely-available comprehensive tool, the *grouped PEPA analyser (GPA)* originally implemented by Stefanek and the results documented in Stefanek et al. [229, 235]. GPA allows ODE-based analysis of many more moments and more complex models than would be possible by hand. It also allows comparisons with simulation, calculation of switch points, and gives access to powerful parameter sweeping and optimisation functionality which can be used to ensure cost-effective satisfaction of service level agreements in massively-parallel models [233].

Chapter 4

Differential-equation approximations of passage times for grouped PEPA

Key contributions
Passage time as time to extinction
Differential-equation analysis of global passage times
Differential-equation analysis of individual passage times

4.1 Introduction

Passage-time or *response-time* distributions are some of the most sought-after quantitative performance measures of a system. Passage-time quantiles form the basis of many service level agreements (SLAs) in the telecommunications and other industries, e.g. a file should be successfully transferred within 2 seconds, 95% of the time. As we have already seen, analysis of such industrial-scale systems requires the ability to deal with massive underlying discrete state spaces which grow exponentially as system components are added to the model. The situation is no better for the computation of passage-time measures. Indeed the capability of traditional explicit state-space techniques for computing passage-time distributions is quickly exceeded [50].

In the last chapter, we have shown how differential-equation based analysis of performance models offers the potential for the analysis of massive state spaces at small computational cost. Specifically, it was shown how the moments of certain discrete stochastic processes which count the number of each type of component in the model in a given state can be approximated by the solutions to tractable systems of ODEs. In this chapter, we show how the quantities obtained in this manner can be used to compute approximations to passage-time distributions. Specifically, we have identified two distinct types of passage-time measure that can be approximated using this approach: *global passage-times*, which capture the evolution of a large component population; and *individual passage-times* which focus on a single tagged component in a model.

We now give a summary of the key contributions of the work presented in this chapter which should also serve as a map of the chapter's structure.

Passage time as time to extinction We build on initial work by Bradley et al. [52] which showed how ODE analysis could be used to extract passage-time measures for a particular class of conditioned random variable in PEPA models.[1] In Section 4.2, we show that a broader class of passage-time measures are equivalent to the *time to extinction* of a certain set of components within a modified model. This perspective allows us to adapt the techniques of the previous chapter to approximate passage-time measures.

Differential-equation analysis of global passage times In Section 4.3, we develop the notion of a *global passage time* as a means of capturing system-wide passages. We show how they can be approximated by ODE-based analysis as a time-to-extinction measure. Specifically, in Section 4.3.1, we introduce a simple deterministic approximation. Where the component populations are not large enough for the deterministic approximation to be accurate, we improve upon this by showing how efficient approximate upper and lower bounds on the cumulative distribution function of the entire passage time can be derived (Section 4.3.2).

Differential-equation analysis of individual passage times Section 4.4 shows how *individual passage times* can track the evolution of single components in massively-parallel systems, both in the steady-state and transient regimes. For these individual passage times the entire cumulative distribution function can be well approximated by ODE-based techniques. Individual passage-time measures are analogous to *tagged-customer* measures in stochastic Petri nets [21, 93]. However the Petri net approaches still rely on traditional explicit-state Markov chain analysis and are thus susceptible to state-space explosion.

In summary, the aim of this chapter is to provide a machinery for the systematic approximation of passage times in performance models with underlying state-space sizes well beyond the capabilities of existing techniques.

4.2 Passage time as time to extinction

In this section we show how certain passage times can be considered as extinction times of a certain set of fluid components. These ideas were inspired by traditional passage-time analysis techniques [14, 94] where absorbing modifications are made to make passage-time measures more explicit. The term *absorbing* usually refers to a single CTMC state from which no other state is reachable. However, we will generalise this notion here to absorbing *sets* of states such that no state outside the set is reachable from a state within the set. We may then approximate the passage-time duration by considering how long it takes for a particular sum of ODE components associated with the absorbing set to decrease to some value.

We will consider again the example client–server GPEPA model $\boldsymbol{CS}(n, m)$ (Section 3.2.3). We may be interested in how long it takes for some proportion of the initial n clients to complete their first cycle (consisting of at least one *request*-, *data*- and *think*-action, perhaps interrupted by *timeout*-actions). As it stands, such a random variable cannot be represented explicitly in

[1] We would like to emphasise that the conditioned passage-times presented in Bradley et al. [52] are distinct from the traditional first passage time measures discussed by Clark [78].

$$\begin{aligned}
\boldsymbol{Client} &\stackrel{def}{=} (request, r_r).\boldsymbol{Client_{waiting}} \\
\boldsymbol{Client_{waiting}} &\stackrel{def}{=} (data, r_d).\boldsymbol{Client_{think}} + (timeout, r_{tmt}).\boldsymbol{Client} \\
\boldsymbol{Client_{think}} &\stackrel{def}{=} (think, r_t).\boldsymbol{Client'} \\
\boldsymbol{Client'} &\stackrel{def}{=} (request, r_r).\boldsymbol{Client_{waiting}}' \\
\boldsymbol{Client_{waiting}}' &\stackrel{def}{=} (data, r_d).\boldsymbol{Client_{think}}' + (timeout, r_{tmt}).\boldsymbol{Client'} \\
\boldsymbol{Client_{think}}' &\stackrel{def}{=} (think, r_t).\boldsymbol{Client'}
\end{aligned}$$

Figure 4.1: New definition of the ***Client*** fluid component in the model $\boldsymbol{CS'}(n, m)$.

the aggregated state space; we wish to represent it as the passage from a given source state to a set of target states. In order to do this, we can modify the $\boldsymbol{Client_{think}}$ fluid component and introduce three new model states and corresponding fluid components $\boldsymbol{Client'}$, $\boldsymbol{Client_{waiting}}'$ and $\boldsymbol{Client_{think}}'$ as specified in Figure 4.1.

These additional states will allow us to distinguish between components which have completed a cycle and ones that have not. The rest of the model is left unchanged. In making this modification and the similar modifications detailed in this chapter, our intention is to preserve the behavioural aspects of the system. We will see in Chapter 5 how this can be achieved more systematically through the use of the *unified stochastic probe* framework. Denote the resulting model:

$$\boldsymbol{CS'}(n, m) \stackrel{def}{=} \mathbf{Clients}\{\boldsymbol{Client}[n]\} \underset{L}{\bowtie} \mathbf{Servers}\{\boldsymbol{Server}[m]\}$$

We are now in a position to express the random variable we are interested in as the time to extinction of the specified number of the ***Client***, $\boldsymbol{Client_{waiting}}$ and $\boldsymbol{Client_{think}}$ components in the modified model. It is easy to see how we could develop a similar modification to, for example, allow us to time how long it takes for a client to complete *any* number of cycles. In the sections that follow, we will show how the systems of ODEs defined in the previous chapter for such modified models could be used to compute approximations to the distribution of passage-time random variables.

As mentioned earlier, we will consider two different classes of passage times both of which are particularly amenable to accurate fluid approximation under the right conditions. Passage times of the first type are called *global passage times*. These passage times represent the time taken for a significant proportion of a component population to reach some state, or achieve some particular goal. The second type is called an *individual passage time*. These will be marginal passage times for individuals in a large population of identically-distributed components.

4.3 Global passage times

We consider again instances of the model introduced above, with even numbers of clients, that is, $\boldsymbol{CS'}(2n, m)$. Consider the passage-time quantity for half (or n) of the clients to complete their initial cycle. As mentioned above, the ODEs of Definition 3.3.1 can be constructed for this model yielding a system of 9 ODEs, which is given in Appendix B.1.1 and is also available

online [131]. In contrast to the case of the unmodified model, $\boldsymbol{CS}(2n, m)$, these ODEs allow us access to the random variable we are interested in. Specifically, it would seem sensible to construct the approximation by considering the deterministic quantity $v_{\boldsymbol{C}}(t) + v_{\boldsymbol{Cw}}(t) + v_{\boldsymbol{Ct}}(t)$ (or $v_{\boldsymbol{C'}}(t) + v_{\boldsymbol{Cw'}}(t) + v_{\boldsymbol{Ct'}}(t)$) and computing the time t at which it reaches the value n. We will present two possible approaches: the first, which yields a deterministic approximation, and the second, which yields approximations to upper and lower bounds on the entire distribution of the passage time. We first define the general class of global passage times which we will be interested in for ODE-based analysis.

A *global passage time* consists of a GPEPA model together with an absorbing subset of its aggregated state space, specified by a particular subset of fluid components and a target count for these fluid components to reach. The passage time is then the time taken for this to occur. The formal definition follows.

Definition 4.3.1 (Global passage time)**.** *Let G be a GPEPA model, $\mathcal{C} \subseteq \mathcal{B}(G)$ be a subset of fluid components and $C \in \mathbb{Z}_+$ represent the target component count. Define the global passage-time random variable $\sigma := \inf\{t \in \mathbb{R}_+ : \sum_{B \in \mathcal{C}} N_B(t) \leq C\}$, where whenever $t > \sigma$, $\sum_{B \in \mathcal{C}} N_B(t) \leq C$, that is, the target states must be absorbing and the passage is timed starting from the initial state of G.*

This definition can be used to describe the example passage time for half of the clients to complete their first cycle by letting $G = \boldsymbol{CS'}(2n, m)$, $\mathcal{C} = \{\boldsymbol{Client}, \boldsymbol{Client_{waiting}}, \boldsymbol{Client_{think}}\}$ and $C = n$.

4.3.1 Approximation by point mass

The most straightforward approach to approximating the passage time mentioned above would be to compute the time t at which the quantity $v_{\boldsymbol{C}}(t)+v_{\boldsymbol{Cw}}(t)+v_{\boldsymbol{Ct}}(t)$ reaches n. Figure 4.2 shows cumulative distribution functions (CDFs) computed using stochastic simulation for this passage-time random variable. In each case, we increase the number of clients and there are always three fifths as many servers as there are clients. Maintaining a constant ratio ensures that the point-mass approximation for each of these passage times is the same (Lemma 3.3.2), represented by the dashed vertical line in the figure. We see that the passage-time distributions appear to be converging to the point mass as the component populations increase (see Section 6.4.1 for general theoretical justification of this fact).

For a general global passage time specified for some GPEPA model G by $\mathcal{C} \subseteq \mathcal{B}(G)$ and $C \in \mathbb{Z}_+$, as in Definition 4.3.1, the point-mass approximation is defined simply as $\inf\{t \in \mathbb{R}_+ : \sum_{B \in \mathcal{C}} v_B(t) \leq C\}$.

4.3.2 Upper and lower CDF approximations

For smaller populations sizes, many of the passage-time distributions depicted in Figure 4.2 have a significant level of variability. In such cases, a deterministic passage-time approximation does not capture an accurate picture. In this section, we show how we might address this by introducing what we term *upper* and *lower* approximations to global passage-time CDFs.

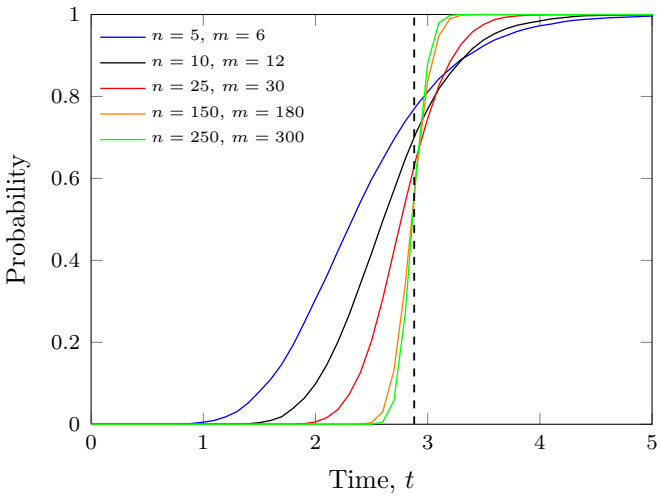


Figure 4.2: Passage-time CDFs for half of the clients to complete their first cycle computed by stochastic simulation compared with the ODE point-mass approximation (the dashed vertical line) for the model $\boldsymbol{CS}(2n, m)$. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.5$, $r_b = 0.05$, $r_d = 2.0$, $r_{rst} = 1.0$.

These are ODE-computed approximations to theoretically exact *upper* and *lower* CDF bounds. Although we do not yet have a formal guarantee or condition to ensure that such approximations are *themselves* bounds on the CDF, we have observed empirically that these approximations are usually very accurate and converge much faster than those of Section 4.3.1. Furthermore, we have not encountered any cases where, under the approximation, the bounding property is violated. However we do note that, in some cases, the additional accuracy comes at the expense of bounds which may not be as tight as desired, although we emphasise here that any CDF lower bound can potentially be used to validate an SLA.

In the next section we will introduce the simplest such approximations computed using only the first-moment ODE approximations for GPEPA models which were given in Section 3.3. Then we will develop tighter approximations by employing the higher-moment approximations of Section 3.4.

First-moment CDF approximations

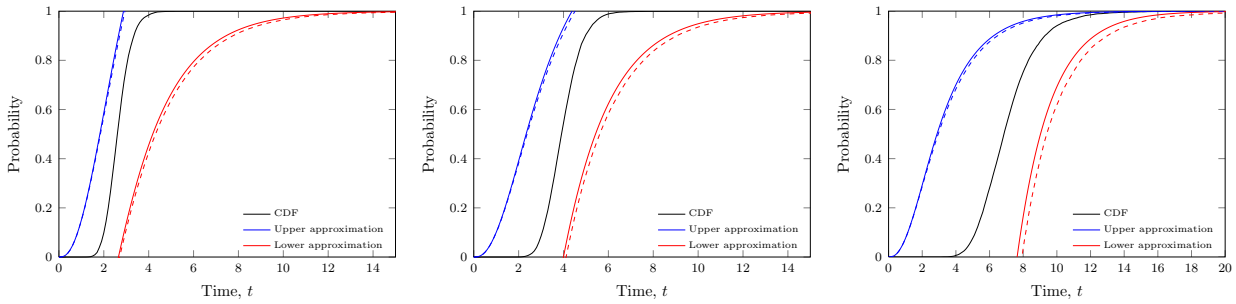
In this section, we will approximate global passage-time distributions by employing the well-known Markov inequality, which says that for a non-negative random variable X and $a > 0$:

$$\mathbb{P}\{X \geq a\} \leq \frac{\mathbb{E}[X]}{a} \tag{4.1}$$

In order to exploit this, we can use the system of ODEs defined in Section 3.3 to approximate the component-count expectations.

Consider again the model $\boldsymbol{CS}'(2n, m)$ and the passage time for n of the clients to complete their first cycle. Denote this random variable by σ again, then applying Markov's inequality, we may obtain the following bounds on its cumulative distribution function:

$$\mathbb{P}\{\sigma \leq t\} = \mathbb{P}\{N_{\boldsymbol{C}'}(t) + N_{\boldsymbol{Cw}'}(t) + N_{\boldsymbol{Ct}'}(t) \geq n\} \leq \frac{\mathbb{E}[N_{\boldsymbol{C}'}(t)] + \mathbb{E}[N_{\boldsymbol{Cw}'}(t)] + \mathbb{E}[N_{\boldsymbol{Ct}'}(t)]}{n}$$



(a) Half of the clients (b) Three quarters of the clients (c) All of the clients

Figure 4.3: Passage-time CDFs for varying proportions of the clients to complete their first cycle computed by stochastic simulation compared with the ODE-derived upper and lower approximation from Markov's inequality for the model $\boldsymbol{CS}(20, 12)$. The dashed line represents the actual theoretical bound, again computed by stochastic simulation. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.5$, $r_b = 0.05$, $r_d = 2.0$, $r_{rst} = 1.0$.

and, since $N_{\boldsymbol{C}}(t) + N_{\boldsymbol{Cw}}(t) + N_{\boldsymbol{Ct}}(t) + N_{\boldsymbol{C'}}(t) + N_{\boldsymbol{Cw'}}(t) + N_{\boldsymbol{Ct'}}(t) = 2n$, also:

$$\begin{aligned} \mathbb{P}\{\sigma \leq t\} &= 1 - \mathbb{P}\{N_{\boldsymbol{C}}(t) + N_{\boldsymbol{Cw}}(t) + N_{\boldsymbol{Ct}}(t) \geq n + 1\} \\ &\geq 1 - \frac{\mathbb{E}[N_{\boldsymbol{C}}(t)] + \mathbb{E}[N_{\boldsymbol{Cw}}(t)] + \mathbb{E}[N_{\boldsymbol{Ct}}(t)]}{n+1} \end{aligned} \tag{4.2}$$

Applying the approximation $\mathbb{E}[\mathbf{N}(t)] \approx \mathbf{v}(t)$ allows us to estimate these bounds using the solutions to the corresponding system of differential equations.

Figure 4.3 gives the results of applying this approach for three different passage-time random variables. In all three cases we note that the ODE approximation to the theoretical bound is generally very accurate. We would certainly expect a more accurate result than the point-mass approximation depicted in Figure 4.2 since that required the entire distribution to concentrate around a point mass. In this section, however, we require only a convergence of expectations, therefore, it would seem that the approximations of this section will be more useful for smaller component populations than the point-mass approximation. On the other hand, as the component populations increase, and the CDFs approach the point-mass approximation, the bounds of this section will necessarily become quite loose. Therefore, at some point, it is certainly likely to be advantageous to switch to the point-mass approximation. Indeed, when only first moments are considered, we would not expect that the bounds would become tighter in the limit of large populations since no measure of variability is considered.

Another interesting point to note in Figure 4.3 is that the relative tightness of the more useful lower bound appears to be increasing in the higher and arguably more useful quantiles as the proportion being timed increases, whereas the upper bound becomes looser everywhere. To see why this is to be expected let σ_a be the passage-time random variable for a of the clients to complete their first cycle. Applying analogous reasoning as for Equation 4.2, we may obtain:

$$\mathbb{E}[N_{\boldsymbol{C}}(t)] + \mathbb{E}[N_{\boldsymbol{Cw}}(t)] + \mathbb{E}[N_{\boldsymbol{Ct}}(t)] \geq \mathbb{P}\{\sigma_a > t\} \times (2n - a + 1)$$

This inequality can also be derived by considering the events $\{\sigma_a > t\}$ and $\{\sigma_a \leq t\}$ and the smallest possible values of $N_{\boldsymbol{C}}(t) + N_{\boldsymbol{Cw}}(t) + N_{\boldsymbol{Ct}}(t)$ on each event. If $\sigma_a \leq t$, the passage

has completed and without extra information, we cannot say anything more than the trivial statement $N_{\boldsymbol{C}}(t) + N_{\boldsymbol{Cw}}(t) + N_{\boldsymbol{Ct}}(t) \geq 0$. In general, after the passage has completed, $N_{\boldsymbol{C}}(t) + N_{\boldsymbol{Cw}}(t) + N_{\boldsymbol{Ct}}(t)$ will not actually be zero and could in fact be as large as $2n - a$. However as a approaches $2n$ the potential for this discrepancy decreases hence we would expect the bound to become tighter as we observe in Figure 4.3. A similar argument can be made for the increasing looseness of the upper approximation.

First-moment CDF approximations for general global passage times

We now show how to bound global passage-time random variables in general. Let σ be a global passage-time random variable specified for some GPEPA model G by $\mathcal{C} \in \mathcal{B}(G)$ and $C \in \mathbb{Z}_+$, as in Definition 4.3.1. We may then compute, writing $\mathcal{H} := \{H : (H, P) \in \mathcal{C}\}$ for the component groups involved in the specification of the global passage time:

$$\begin{aligned}
\mathbb{P}\{\sigma \leq t\} &= \mathbb{P}\left\{\sum_{B \in \mathcal{C}} N_B(t) \leq C\right\} \\
&= \mathbb{P}\left\{\sum_{H_i \in \mathcal{H}} \sum_{(H_i, P_j) \in \mathcal{C}} N_{H_i,P_j}(t) \leq C\right\} \\
&= \mathbb{P}\left\{\sum_{H_i \in \mathcal{H}} \sum_{(H_i, P_j) \in \mathcal{C}} N_{H_i,P_j}(t) - \sum_{H_i \in \mathcal{H}} \mathcal{S}(G, H_i) \leq C - \sum_{H_i \in \mathcal{H}} \mathcal{S}(G, H_i)\right\} \\
&= \mathbb{P}\left\{\sum_{H_i \in \mathcal{H}} \left(\mathcal{S}(G, H_i) - \sum_{(H_i, P_j) \in \mathcal{C}} N_{H_i,P_j}(t)\right) \geq \sum_{H_i \in \mathcal{H}} \mathcal{S}(G, H_i) - C\right\}
\end{aligned}$$

Since for any passage-time random variable not identically zero, the right-hand side of the above is strictly positive, we may apply Markov's inequality to obtain:

$$\mathbb{P}\{\sigma \leq t\} \leq \frac{\sum_{H_i \in \mathcal{H}} (\mathcal{S}(G, H_i) - \sum_{(H_i, P_j) \in \mathcal{C}} \mathbb{E}[N_{H_i,P_j}(t)])}{\sum_{H_i \in \mathcal{H}} \mathcal{S}(G, H_i) - C} \tag{4.3}$$

Working in the other direction, we have:

$$\mathbb{P}\{\sigma \leq t\} = 1 - \mathbb{P}\left\{\sum_{B \in \mathcal{C}} N_B(t) > C\right\} = 1 - \mathbb{P}\left\{\sum_{B \in \mathcal{C}} N_B(t) \geq C + 1\right\} \tag{4.4}$$

Applying Markov's inequality directly, we obtain:

$$\mathbb{P}\{\sigma \leq t\} \geq 1 - \frac{1}{C+1} \sum_{(H,P) \in \mathcal{C}} \mathbb{E}[N_{H,P}(t)] \tag{4.5}$$

The approximation $\mathbb{E}[\mathbf{N}(t)] \approx \mathbf{v}(t)$ can then be applied directly in either case to provide the bound estimates.

Higher-moment CDF approximations

In this section, we show how the techniques of the last section can be improved by exploiting the differential-equation approximations to higher-order moments as constructed in Section 3.4.3.

We will begin by replacing the use of Markov's inequality in the previous section with Chebyshev's inequality. If X is an arbitrary random variable, $t > 0$ and $q \neq 0$, Chebyshev's inequality says [e.g. 221, Theorem A.113, Page 492]:

$$\mathbb{P}\{|X - \mathbb{E}[X]| \geq b\} \leq \frac{\mathbb{E}[|X - \mathbb{E}[X]|^q]}{b^q}$$

For now, we will consider the case $q = 2$ which introduces variance information. This should result in bounds that are tighter than those of the last section. In the case $q = 2$, it is a straightforward application of the Cauchy–Schwarz inequality to derive a one-sided refinement of this, often known as the Chebyshev–Cantelli inequality, which says that for $b > 0$:

$$\mathbb{P}\{X - \mathbb{E}[X] \geq b\} \leq \frac{\mathrm{Var}[X]}{\mathrm{Var}[X] + b^2} \qquad \mathbb{P}\{\mathbb{E}[X] - X \geq b\} \leq \frac{\mathrm{Var}[X]}{\mathrm{Var}[X] + b^2}$$

As before, let σ be a global passage-time random variable specified for some GPEPA model G by $\mathcal{C} \in \mathcal{B}(G)$ and $C \in \mathbb{Z}_+$, as in Definition 4.3.1. Then we have:

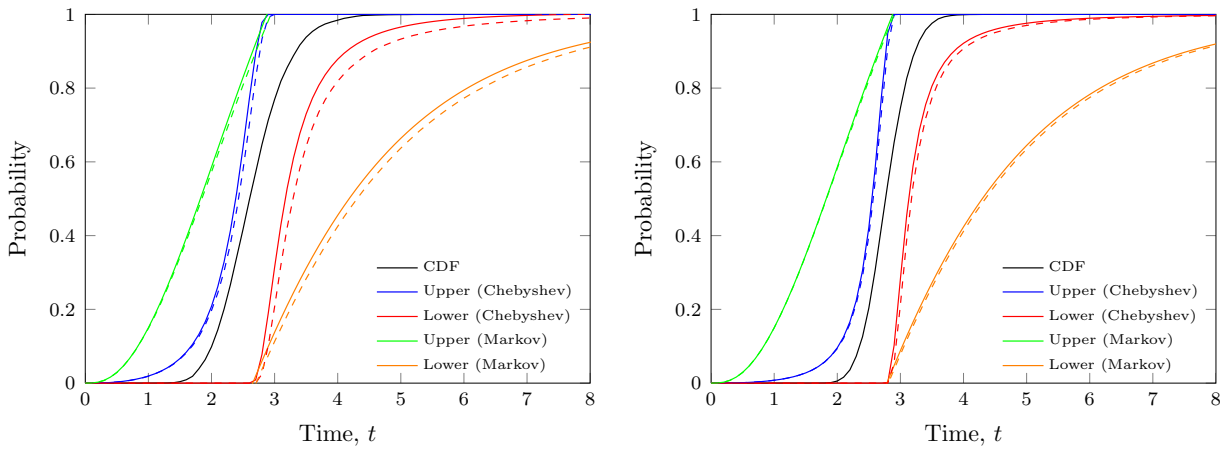
$$\begin{aligned}\mathbb{P}\{\sigma \leq t\} = \mathbb{P}\left\{\sum_{B\in\mathcal{C}} N_B(t) \leq C\right\} &= \mathbb{P}\left\{\sum_{B\in\mathcal{C}} (\mathbb{E}[N_B(t)] - N_B(t)) \geq \sum_{B\in\mathcal{C}} \mathbb{E}[N_B(t)] - C\right\} \\ &\leq \frac{\mathrm{Var}\left[\sum_{B\in\mathcal{C}} N_B(t)\right]}{\mathrm{Var}\left[\sum_{B\in\mathcal{C}} N_B(t)\right] + (\mathbb{E}\left[\sum_{B\in\mathcal{C}} N_B(t)\right] - C)^2} \qquad (4.6)\end{aligned}$$

where the inequality is valid when $\mathbb{E}\left[\sum_{B\in\mathcal{C}} N_B(t)\right] - C > 0$. Similarly, working in the other direction, we may obtain:

$$\begin{aligned}\mathbb{P}\{\sigma \leq t\} &= 1 - \mathbb{P}\left\{\sum_{B\in\mathcal{C}} N_B(t) \geq C + 1\right\} \qquad (4.7) \\ &= 1 - \mathbb{P}\left\{\sum_{B\in\mathcal{C}} (N_B(t) - \mathbb{E}[N_B(t)]) \geq C + 1 - \sum_{B\in\mathcal{C}} \mathbb{E}[N_B(t)]\right\} \\ &\geq 1 - \frac{\mathrm{Var}\left[\sum_{B\in\mathcal{C}} N_B(t)\right]}{\mathrm{Var}\left[\sum_{B\in\mathcal{C}} N_B(t)\right] + (\mathbb{E}[\sum_{B\in\mathcal{C}} N_B(t)] - C - 1)^2} \qquad (4.8)\end{aligned}$$

where the inequality is valid when $C + 1 - \mathbb{E}[\sum_{B\in\mathcal{C}} N_B(t)] > 0$. We may then apply the differential-equation approximations to first- and second-order moments defined in Sections 3.3 and 3.4 to yield approximations to these bounds. Figure 4.4 shows how these bounds can substantially improve on those obtained using Markov's inequality. We also observe that they appear to increase substantially in tightness as the component populations increase. The second-moment ODEs for $\boldsymbol{CS}'(2n, n)$ used to generate this figure are available online [131].

The first- and second-order CDF approximations discussed above are obtained by addressing the general problem in probability theory of retrieving a distribution from a selection of its moments, the so-called *reduced moment problem*. A more advanced technique is described



(a) $n = 10$, $m = 12$ (b) $n = 25$, $m = 30$

Figure 4.4: Passage-time CDFs for half of the clients to complete their first cycle computed by stochastic simulation compared with the ODE-derived upper and lower approximation from Chebyshev's inequality for the model $\boldsymbol{CS}(2n, m)$. The dashed line represents the actual theoretical bound, again computed by stochastic simulation. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.5$, $r_b = 0.05$, $r_d = 2.0$, $r_{rst} = 1.0$.

by Tari et al. [241] that can utilise moments of orders higher than two. Although application of these techniques are beyond the scope of this thesis, they have been applied in Hayden et al. [142] to obtain tighter global passage-time approximations by utilising ODE-derived moment approximations of up to fourth order.

Finally, we note that for global passage times timing the complete extinction of a population (such as that depicted in Figure 4.3c), the point-mass approximation of Section 4.3.1 cannot be applied directly since the relevant ODE quantity never actually reaches zero. In these cases, then, we would advise the practitioner to use the upper and lower CDF approximation techniques of this section.

4.4 Individual passage times

In this section we will consider the ODE-based approximation of a second kind of passage-time measurement, the *individual passage time*. These are marginal passage times for individuals (fluid components) which are part of a large population of similar fluid components. We will show how the entire cumulative distribution function of such random variables can be approximated directly. This builds on approaches described by Clark et al. [81], Ding [91], Massoulié and Vojnovic [185], Shakkottai and Johari [220], Tribastone [249], amongst others, which are able to obtain *average* individual passage-time measures by combining ODE-based techniques and Little's Law [181].

By means of introduction, consider again the model $\boldsymbol{CS}'(n, m)$ and in contrast to the last section, assume now that we are interested in how long it takes for one of the initial n clients to complete its first cycle. Since all members of the **Clients** component group are identically

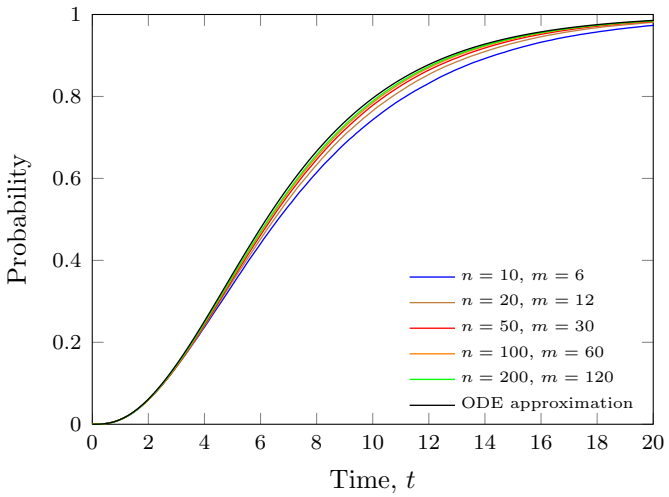


Figure 4.5: Passage-time CDFs for a single client to complete its first cycle computed by stochastic simulation compared with the ODE-derived approximation for the model $\boldsymbol{CS}(n, m)$. Rates: $r_r = 1.0$, $r_{tmt} = 0.1$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 0.7$, $r_{rst} = 0.5$.

distributed it makes sense to speak of this passage-time measurement independent of any specific individual. Let $C_j(t) \in ds(\boldsymbol{Client})$ for $1 \leq j \leq n$ be the stochastic process which tracks the state of the jth individual in the $\boldsymbol{Client}$ component group. We wish to evaluate at time $t \in \mathbb{R}_+$ and for any $1 \leq k \leq n$, the cumulative distribution function:

$$\mathbb{P}\{C_k(t) \in \{\boldsymbol{Client}', \boldsymbol{Client_{waiting}}', \boldsymbol{Client_{think}}'\}\}$$

That is, the probability that by time t, a specific individual client has completed its first cycle. Now we have that:

$$\mathbb{E}[N_{\boldsymbol{C}'}(t) + N_{\boldsymbol{Cw}'}(t) + N_{\boldsymbol{Ct}'}(t)] = \sum_{j=1}^{n} \mathbb{P}\{C_j(t) \in \{\boldsymbol{Client}', \boldsymbol{Client_{waiting}}', \boldsymbol{Client_{think}}'\}\}$$

So, by the fact that the $C_j(t)$ are identically distributed, we can compute the quantity of interest as:

$$\mathbb{P}\{C_k(t) \in \{\boldsymbol{Client}', \boldsymbol{Client_{waiting}}', \boldsymbol{Client_{think}}'\}\} = \frac{1}{n}(\mathbb{E}[N_{\boldsymbol{C}'}(t)] + \mathbb{E}[N_{\boldsymbol{Cw}'}(t)] + \mathbb{E}[N_{\boldsymbol{Ct}'}(t)])$$

If we again make the approximation $\mathbb{E}[\mathbf{N}(t)] \approx \mathbf{v}(t)$ this provides the route to the ODE approximation of the entire cumulative distribution function of the individual passage time.

Figure 4.5 shows cumulative distribution functions computed using traditional methods for the passage-time random variable discussed above. In each case, we increase the number of clients and there are always three fifths as many servers as there are clients. This ensures, similarly to the previous section, that the differential-equation approximation to the distribution for each of these passage times is identical (Lemma 3.3.2). We see that the cumulative distribution functions do appear to be converging to the ODE approximation as the component populations increase.

The example individual passage time we have just given above is an instance of the very simple class of individual passage-time measures which are timed starting immediately from

the model's initial state and end when the fluid component under observation enters some absorbing subset of *target states*. In the next two sections, we will show how more general and useful individual passage-time measurements can be computed using ODE-based techniques. Specifically, in the next section we introduce and discuss individual passage times measured in the steady-state regime. Then we will look at more general transient individual passage times where the timing only starts when the fluid component under observation enters a state in some set of *start states*.

4.4.1 Steady-state individual passage times

As opposed to timing a component from a model's fixed initial state, we will often be interested in measuring the time taken for a fluid component under observation to move from one of a set of designated *start states* to a *target state* assuming that the model is operating in its steady-state or stationary regime. In order to compute such passage times, it is necessary to consider two GPEPA models — one that is used to compute the model's stationary distribution; and one where the fluid component under observation has a set of local derivative states which are absorbing (the target states) that is used to measure the passage time. In fact, using the CTMC stationary distribution for this purpose is actually an approximation. Instead, it is the stationary distribution of an appropriate *embedded discrete-time Markov chain* which should be used here in order to capture the model state at jump instants. We discuss this issue in more detail later in Section 4.4.3.

We will consider again the client–server model introduced in Section 3.2.3. Assume we are interested in the same passage time as in the previous section — the time taken for a client to complete one cycle — but measured when the model is operating in the steady-state regime. As mentioned above, we will need to use both the unmodified client–server GPEPA model $\boldsymbol{CS}(n,m)$ and the version with the absorbing modification $\boldsymbol{CS}'(n,m)$. The model $\boldsymbol{CS}(n,m)$ will be used to compute the stationary distribution with which we will initialise the model $\boldsymbol{CS}'(n,m)$ in order to compute the actual passage-time distribution.

Approximation of stationary expectations of $\boldsymbol{CS}(n,m)$

The first stage in the computation of passage times of this form is to approximate the stationary expectations of the fluid component counts of $\boldsymbol{CS}(n,m)$ using its approximating system of ODEs given by Equation 3.4. Since we are interested in the stationary expectations, it makes sense to take the limit of the ODE solutions as $t \to \infty$. To compute these quantities, one can either integrate the ODEs numerically for a sufficiently large period of time or attempt to find a unique meaningful fixed point of the ODEs[2] by solving the system of algebraic equations obtained when the right-hand sides of the ODEs are equated with zero. We will write for example $v_{\boldsymbol{C}}$ for the ODE approximation to the stationary expectation $N_{\boldsymbol{C}} := \lim_{t\to\infty} \mathbb{E}[N_{\boldsymbol{C}}(t)]$ and so on for the other fluid components in $\boldsymbol{CS}(n,m)$. We will later write component counts

[2]The system of ODEs usually has infinitely many fixed points, but often has 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 Chapter 6 where conditions guaranteeing convergence of the stationary distribution to the ODE's fixed point for increasing component population size are given.

in the model $\boldsymbol{CS}'(n,m)$ with primes, for example, $N'_{\boldsymbol{C}}(t)$ and $N'_{\boldsymbol{C}'}(t)$; and similarly for their first-moment ODE approximations $v'_{\boldsymbol{C}}(t)$ and $v'_{\boldsymbol{C}'}(t)$, respectively.

The approximation of the passage-time CDF will then be obtained by solving the system of ODEs corresponding to $\boldsymbol{CS}'(n,m)$, initialised with the stationary expectations obtained above. More specifically, consider the model $\boldsymbol{CS}(n,m)$ evolving in its stationary regime and then at some fixed time switch its evolution to that of $\boldsymbol{CS}'(n,m)$, initialised by the current state of $\boldsymbol{CS}(n,m)$. The idea is then similar to that of the last section: by timing how long it takes for a client in the $\boldsymbol{Client}$ state to reach one of the target states in $\mathcal{T} := \{\boldsymbol{Client}', \boldsymbol{Client_{waiting}}', \boldsymbol{Client_{think}}'\}$ we can measure the duration of the passage.

Approximation of the passage-time CDF

To see how we proceed, define $C_j(t) \in ds(\boldsymbol{Client})$ to be the stochastic process, which tracks the state of the jth fluid component in the group **Clients** as the model $\boldsymbol{CS}(n,m)$ evolves. Consider the model $\boldsymbol{CS}(n,m)$ evolving in its stationary regime. Let time 0 be the arbitrary fixed time at which the evolution switches to $\boldsymbol{CS}'(n,m)$, initialised by the state of $\boldsymbol{CS}(n,m)$. Then the quantity we wish to compute is:[3]

$$\mathbb{P}\{C_j(t) \in \mathcal{T} \mid C_j(0) = \boldsymbol{Client}\} \tag{4.9}$$

Similarly to the previous section, we may use the identically-distributed nature of the **Clients** component group to compute $\mathbb{P}\{C_j(0) = \boldsymbol{Client}\} = \frac{1}{n}\mathbb{E}[N_{\boldsymbol{C}}(0)] = \frac{1}{n}N_{\boldsymbol{C}}$. Therefore, in order to compute the conditional expression of Equation 4.9, it suffices to find:

$$\mathbb{P}(\{C_j(t) \in \mathcal{T}\} \cap \{C_j(0) = \boldsymbol{Client}\}) \tag{4.10}$$

It turns out that we can gain access to this quantity in terms of the component-count expectations of $\boldsymbol{CS}'(n,m)$. However, this requires that we initialise the fluid component counts in $\boldsymbol{CS}'(n,m)$ slightly differently at time 0 than may initially be expected:

$$\begin{array}{lll}
N'_{\boldsymbol{C}}(0) := N_{\boldsymbol{C}}(0) & N'_{\boldsymbol{C}'}(0) := 0 & N'_{\boldsymbol{S}}(0) := N_{\boldsymbol{S}}(0) \\
N'_{\boldsymbol{Cw}}(0) := 0 & N'_{\boldsymbol{Cw}'}(0) := N_{\boldsymbol{Cw}}(0) & N'_{\boldsymbol{Sg}}(0) := N_{\boldsymbol{Sg}}(0) \\
N'_{\boldsymbol{Ct}}(0) := 0 & N'_{\boldsymbol{Ct}'}(0) := N_{\boldsymbol{Ct}}(0) & N'_{\boldsymbol{Sb}}(0) := N_{\boldsymbol{Sb}}(0)
\end{array}$$

Specifically, we do not initialise $N'_{\boldsymbol{Cw}}(0)$ with $N_{\boldsymbol{Cw}}(0)$, or $N'_{\boldsymbol{Ct}}(0)$ with $N_{\boldsymbol{Ct}}(0)$, but instead we place such client component counts in their equivalent absorbing states in $\boldsymbol{CS}'(n,m)$. This does not affect the value of Equation 4.10 but does allow us to single out the client fluid components which were at the start of a passage (in state $\boldsymbol{Client}$) at time 0 and track their evolution through to time t. In this way we now have that:[4]

$$\sum_{Q \in \mathcal{T}} \left(N'_Q(t) - N'_Q(0)\right) = \sum_{j=1}^{n} \mathbf{1}_{\{C_j(t) \in \mathcal{T}\} \cap \{C_j(0) = \boldsymbol{Client}\}}$$

[3] In fact, the exact quantity of interest is $\mathbb{P}\{C_j(t) \in \mathcal{T} \mid C_j(0) = \boldsymbol{Client}, C_j(0-) = \boldsymbol{Client_{think}}\}$. However, for this example, this additional approximation does not affect the asymptotic correctness of the ODE approximation. This issue is discussed in more detail and generality in Section 4.4.3.

[4] Where $\mathbf{1}_A$ is the indicator function of the event A.

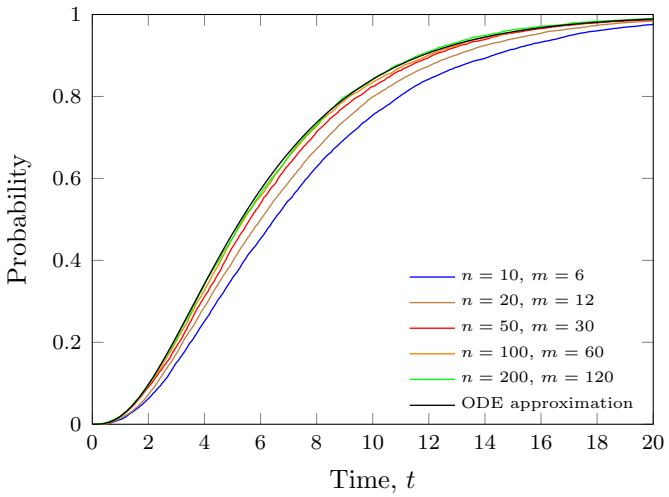


Figure 4.6: Passage-time CDFs for a single client to complete a cycle in the steady-state regime computed by stochastic simulation compared with the ODE-derived approximation for the model $\boldsymbol{CS}(n, m)$. Rates: $r_r = 1.0$, $r_{tmt} = 0.1$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 0.7$, $r_{rst} = 0.5$.

Thus on taking expectations and dividing by $\mathbb{P}\{C_j(0) = \boldsymbol{Client}\} = \frac{1}{n}\mathbb{E}[N_{\boldsymbol{C}}(0)]$, we may obtain:

$$\mathbb{P}\{C_j(t) \in \mathcal{T} \mid C_j(0) = \boldsymbol{Client}\} = \frac{\sum_{Q \in \mathcal{T}} \left(\mathbb{E}[N'_Q(t)] - \mathbb{E}[N'_Q(0)]\right)}{\mathbb{E}[N_{\boldsymbol{C}}(0)]} \tag{4.11}$$

We now aim to replace the expectation terms in the above by the appropriate ODE approximation. We need to specify the initial conditions for the ODEs associated to the modified absorbing model $\boldsymbol{CS}'(n, m)$ at time 0 in terms of the long-time limits of the ODEs from the original model $\boldsymbol{CS}(n, m)$. This must happen according to the preceding discussion regarding how the fluid component counts of $\boldsymbol{CS}'(n, m)$ are initialised by those of $\boldsymbol{CS}(n, m)$ at time 0:

$$\begin{array}{lll} v'_{\boldsymbol{C}}(0) := v_{\boldsymbol{C}} & v'_{\boldsymbol{C}'}(0) := 0 & v'_{\boldsymbol{S}}(0) := v_{\boldsymbol{S}} \\ v'_{\boldsymbol{Cw}}(0) := 0 & v'_{\boldsymbol{Cw}'}(0) := v_{\boldsymbol{Cw}} & v'_{\boldsymbol{Sg}}(0) := v_{\boldsymbol{Sg}} \\ v'_{\boldsymbol{Ct}}(0) := 0 & v'_{\boldsymbol{Ct}'}(0) := v_{\boldsymbol{Ct}} & v'_{\boldsymbol{Sb}}(0) := v_{\boldsymbol{Sb}} \end{array}$$

Then the fluid approximation for the CDF is given by:

$$\mathbb{P}\{C_j(t) \in \mathcal{T} \mid C_j(0) = \boldsymbol{Client}\} \approx \frac{\sum_{Q \in \mathcal{T}} \left(v'_Q(t) - v'_Q(0)\right)}{v_{\boldsymbol{C}}}$$

Figure 4.6 shows cumulative distribution functions computed using traditional methods for the passage-time random variable discussed above. We see that the cumulative distribution functions do appear to be converging to the ODE approximation as the component populations increase (see Section 6.4.2 for general theoretical justification of this fact).

Giving a general presentation of this analysis for steady-state individual passage times on arbitrary models would be unwieldy without a better means to specify measurements and the required local state-space transformations. For this reason, we defer presenting this analysis methodology for general steady-state individual passage times until Section 5.5.1 of the next chapter where we have the power of the *unified stochastic probe* framework at our disposal.

$$
\begin{aligned}
\boldsymbol{Client} &\stackrel{def}{=} (request, r_r).\boldsymbol{Client_{waiting}} \\
\boldsymbol{Client_{waiting}} &\stackrel{def}{=} (data, r_d).\boldsymbol{Client_{think}} + (timeout, r_{tmt}).\boldsymbol{Client} \\
\boldsymbol{Client_{think}} &\stackrel{def}{=} (think, r_t).\widehat{\boldsymbol{Client}} \\
\widehat{\boldsymbol{Client}} &\stackrel{def}{=} (request, r_r).\widehat{\boldsymbol{Client_{waiting}}} \\
\widehat{\boldsymbol{Client_{waiting}}} &\stackrel{def}{=} (data, r_d).\widehat{\boldsymbol{Client_{think}}} + (timeout, r_{tmt}).\widehat{\boldsymbol{Client}} \\
\widehat{\boldsymbol{Client_{think}}} &\stackrel{def}{=} (think, r_t).\boldsymbol{Client'} \\
\boldsymbol{Client'} &\stackrel{def}{=} (request, r_r).\boldsymbol{Client_{waiting}}' \\
\boldsymbol{Client_{waiting}}' &\stackrel{def}{=} (data, r_d).\boldsymbol{Client_{think}}' + (timeout, r_{tmt}).\boldsymbol{Client'} \\
\boldsymbol{Client_{think}}' &\stackrel{def}{=} (think, r_t).\boldsymbol{Client'}
\end{aligned}
$$

Figure 4.7: New definition of the ***Client*** fluid component in the model $\boldsymbol{CS}_2'(n, m)$.

4.4.2 Transient individual passage times

The final type of individual passage time we will consider is a generalisation of the transient individual passage time considered at the beginning of Section 4.4. Instead of starting to time the passage immediately, we will start timing when the fluid component under observation enters one of a set of designated *start states*.

Consider again the client–server model introduced in Section 3.2.3. Earlier, we showed how to compute the time taken for a client to complete one cycle, measured from the initial state of the model. Now assume that instead we wish to measure the time taken to complete the *second cycle*. This requires that we start timing when the first cycle finishes. In order for this to be possible, we must, similarly to preceding sections, modify the model $\boldsymbol{CS}(n, m)$ so that the completion of two cycles is represented explicitly in the state space. As before, this necessitates a modification of the ***Client*** fluid component as specified in Figure 4.7 and we call the version of the client–server model defined using this client component $\boldsymbol{CS}_2'(n, m)$. The system of 12 ODEs corresponding to this model, according to Definition 3.3.1, is given in Appendix B.1.2 and is also available online [131].

As before, since all fluid components in the **Clients** component group are identically distributed, this passage time is the same for any of the components in the **Clients** component group. Then for the fluid component under observation, define random variables S and T for the times at which it enters the $\widehat{\boldsymbol{Client}}$ state and the ***Client'*** state, respectively. The quantity we seek is the distribution of the random variable $T - S$, that is, the duration of the client's passage between the $\widehat{\boldsymbol{Client}}$ and ***Client'*** states.

To proceed, let $F(t) := \mathbb{P}\{T - S \leq t\}$ be the cumulative distribution function of interest, let $F_s(t) := \mathbb{P}\{T - S \leq t \mid S = s\}$ be this CDF conditioned on the passage starting at time s and $g(s) := \frac{\mathrm{d}}{\mathrm{d}s}G(s)$ where $G(s) := \mathbb{P}\{S \leq s\}$. Then we have:

$$F(t) = \int_0^\infty F_s(t)g(s)\,\mathrm{d}s \tag{4.12}$$

In the following sections, we show how to approximate $F_s(t)$ and $g(s)$ in a similar fashion to the approach of Section 4.4.1 using ODE-based techniques. Then applying Equation 4.12 yields the approximation to the CDF of interest.

Approximation of $F_s(t)$

In order to compute $F_s(t)$, we follow a very similar approach to that of Section 4.4.1. We will represent $F_s(t)$ in terms of the component-count expectations, which will again be approximated by the system of ODEs derived from the GPEPA model $\boldsymbol{CS}_2'(n,m)$. We will write component counts in $\boldsymbol{CS}_2'(n,m)$ as, for example, $N_{\boldsymbol{C}}(t)$ or $N_{\boldsymbol{C}'}(t)$ and corresponding first-moment ODE solutions as $v_{\boldsymbol{C}}(t)$ or $v_{\boldsymbol{C}'}(t)$, respectively, where the initial conditions here are given by the initial state of $\boldsymbol{CS}_2'(n,m)$.

As we have done in preceding sections, define $C_j(t) \in ds(\boldsymbol{Client})$ to be the stochastic process, which tracks the state of the jth fluid component in the group **Clients** as the model $\boldsymbol{CS}_2'(n,m)$ evolves. We may then write for any $1 \leq j \leq n$ and $s,t \geq 0$:[5]

$$F_s(t) = \mathbb{P}\{C_j(s+t) \in \mathcal{T} \mid C_j(s) = \widehat{\boldsymbol{Client}}\} \tag{4.13}$$

where $\mathcal{T} := \{\boldsymbol{Client}', \boldsymbol{Client_{waiting}}', \boldsymbol{Client_{think}}'\}$ is the set of target states. Furthermore, we have that:

$$\mathbb{P}\{C_j(s) = \widehat{\boldsymbol{Client}}\} = \frac{1}{n}\mathbb{E}[N_{\widehat{\boldsymbol{C}}}(s)] \tag{4.14}$$

So to approximate $F_s(t)$ by differential equations, it suffices to compute in terms of the component-count expectations, the quantity:

$$\mathbb{P}(\{C_j(s+t) \in \mathcal{T}\} \cap \{C_j(s) = \widehat{\boldsymbol{Client}}\}) \tag{4.15}$$

In a similar fashion to the approach of Section 4.4.1, we may do this by letting the model $\boldsymbol{CS}_2'(n,m)$ evolve up to time s, at which point we set all of the client components to their equivalent absorbing (primed) states, *except for fluid components at the start of a passage time*, that is, those which are in the $\widehat{\boldsymbol{Client}}$ state. We then allow the model to continue to evolve for the further period of time of length t. Again, this does not affect the value of Equation 4.15 but does allow us to single out the components which were at the start of a passage at time s and track their evolution through to time $s+t$. In this way we now have that:

$$\sum_{Q \in \mathcal{T}} \big(N_Q(s+t) - N_Q(s)\big) = \sum_{j=1}^{n} \mathbf{1}_{\{C_j(s+t) \in \mathcal{T}\} \cap \{C_j(s) = \widehat{\boldsymbol{Client}}\}}$$

Taking expectations and combining with Equation 4.14 then yields:

$$\mathbb{P}\{C_j(s+t) \in \mathcal{T} \mid C_j(s) = \widehat{\boldsymbol{Client}}\} = \frac{\sum_{Q \in \mathcal{T}} \big(\mathbb{E}[N_Q(s+t)] - \mathbb{E}[N_Q(s)]\big)}{\mathbb{E}[N_{\widehat{\boldsymbol{C}}}(s)]} \tag{4.16}$$

Then in order to compute the fluid approximation to $F_s(t)$, we construct a different solution to the system of ODEs derived from $\boldsymbol{CS}_2'(n,m)$. We will write this solution as $v^s_Q(t)$ since

[5] In fact, like before, the correct equation is $F_s(t) = \mathbb{P}\{C_j(s+t) \in \mathcal{T} \mid C_j(s) = \widehat{\boldsymbol{Client}}, C_j(s-) = \boldsymbol{Client_{think}}\}$. However, again for this example, this additional approximation does not affect the asymptotic correctness of the ODE approximation. This issue is discussed in more detail and generality in Section 4.4.3.

its initial conditions will be defined in terms of the quantities $v_Q(s)$. Specifically, the $v^s_Q(t)$ are obtained by integrating the ODEs associated with $\boldsymbol{CS}_2{}'(n,m)$ subject to the particular initial conditions which guarantee that all of the client components are set to their equivalent absorbing states apart from those in the $\widehat{\boldsymbol{Client}}$ state, as described above. Formally, these initial conditions are given by:

$$\begin{array}{llll} v^s_{\boldsymbol{C}}(0) := 0 & v^s_{\widehat{\boldsymbol{C}}}(0) := v_{\widehat{\boldsymbol{C}}}(s) & v^s_{\boldsymbol{C}'}(0) := v_{\boldsymbol{C}}(s) & v^s_{\boldsymbol{S}}(0) := v_{\boldsymbol{S}}(s) \\ v^s_{\boldsymbol{Cw}}(0) := 0 & v^s_{\widehat{\boldsymbol{Cw}}}(0) := 0 & v^s_{\boldsymbol{Cw}'}(0) := v_{\boldsymbol{Cw}}(s) + v_{\widehat{\boldsymbol{Cw}}}(s) & v^s_{\boldsymbol{Sg}}(0) := v_{\boldsymbol{Sg}}(s) \\ v^s_{\boldsymbol{Ct}}(0) := 0 & v^s_{\widehat{\boldsymbol{Ct}}}(0) := 0 & v^s_{\boldsymbol{Ct}'}(0) := v_{\boldsymbol{Ct}}(s) + v_{\widehat{\boldsymbol{Ct}}}(s) & v^s_{\boldsymbol{Sb}}(0) := v_{\boldsymbol{Sb}}(s) \end{array}$$

The approximation to $F_s(t)$ is then:

$$F_s(t) \approx \frac{\sum_{Q \in \mathcal{T}} \left(v^s_Q(t) - v^s_Q(0) \right)}{v_{\widehat{\boldsymbol{C}}}(s)}$$

Approximation of $g(s)$

Recall that $G(s)$ is the CDF of the time it takes for a client to start the passage of interest and $g(s)$ is the corresponding probability density function. Consider the set of all states a client can be in after having started the passage of interest, that is, after having completed its first cycle:

$$\mathcal{Q} := \{\widehat{\boldsymbol{Client}}, \widehat{\boldsymbol{Client}_{\boldsymbol{waiting}}}, \widehat{\boldsymbol{Client}_{\boldsymbol{think}}}, \boldsymbol{Client}', \boldsymbol{Client}_{\boldsymbol{waiting}}{}', \boldsymbol{Client}_{\boldsymbol{think}}{}'\}$$

Then we have, again due to the identical distribution of the fluid components, that, for any $1 \leq j \leq n$:

$$G(s) = \mathbb{P}\{C_j(s) \in \mathcal{Q}\} = \frac{1}{n} \sum_{Q \in \mathcal{Q}} \mathbb{E}[N_Q(s)]$$

Applying the ODE approximation $v_Q(s) \approx \mathbb{E}[N_Q(s)]$ and then differentiating provides the approximation of $g(s)$ as required.[6]

Approximation of the CDF $F(t)$

The cumulative distribution function of interest can then be computed directly from Equation 4.12. Figure 4.8 shows cumulative distribution functions computed using traditional methods for the passage-time random variable discussed above. We see that the cumulative distribution functions do appear to be converging to the ODE approximation as the component populations increase.

As in the last section, giving a general presentation of this analysis on arbitrary models would be unwieldy without a better means to specify measurements and the required local state-space transformations. So we again defer the presentation of the analysis methodology for general transient individual passage times until Section 5.5.2 of the next chapter in the context of the *unified stochastic probe* framework.

[6] It is not actually necessary to numerically differentiate here since $\frac{\mathrm{d}}{\mathrm{d}s} v_Q(s)$ is simply the right-hand side of the respective differential equation and so is already known.

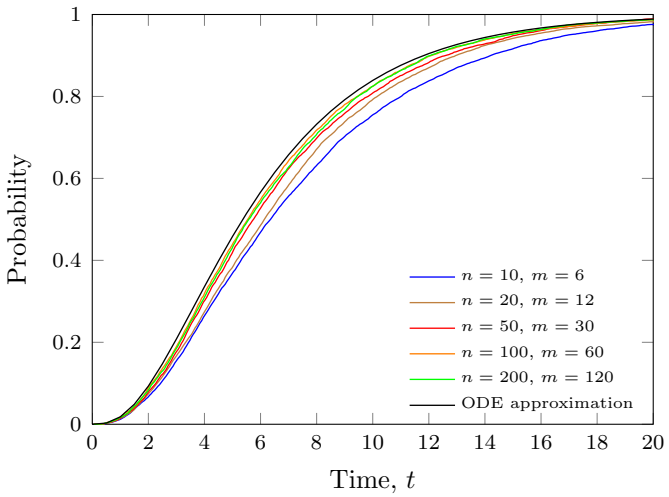


Figure 4.8: Passage-time CDFs for a single client to complete its second cycle computed by stochastic simulation compared with the ODE-derived approximation for the model $\boldsymbol{CS}(n, m)$. Rates: $r_r = 1.0$, $r_{tmt} = 0.1$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 0.7$, $r_{rst} = 0.5$.

4.4.3 Conditioning on transition instants

In the two preceding sections, we have shown how the quantities of Equations 4.9 and 4.13, respectively, can be approximated using differential-equation techniques. However, these expressions do not exactly match the usual passage-time random variable definitions. The exact quantities we are in fact interested in are, respectively:

$$\mathbb{P}\{C_j(t) \in \mathcal{T} \mid C_j(0) = \boldsymbol{Client}, C_j(0-) = \boldsymbol{Client_{think}}\} \tag{4.17}$$

in the steady-state case, and:

$$\mathbb{P}\{C_j(s+t) \in \mathcal{T} \mid C_j(s) = \widehat{\boldsymbol{Client}}, C_j(s-) = \boldsymbol{Client_{think}}\} \tag{4.18}$$

in the transient case.

That is to say that time 0 (respectively, time s) should be the *instant* at which the fluid component under observation performs the *think*-action. Equations 4.9 and 4.13 do not capture this exactly — rather there we are simply guaranteed that at that time, the observed *fluid component* has made no further timed transitions since originally entering the state $\boldsymbol{Client}$ (respectively, $\widehat{\boldsymbol{Client}}$), however, *other components in the model* may well have done.

In order to compute the quantities of Equations 4.17 and 4.18 exactly, it is necessary to consider the *embedded discrete-time subchain* of the original CTMC which is obtained by sampling the CTMC immediately after one of the selected transitions fires. Due to the fact that the client–server passage-time examples considered here have only one *local* source state ($\boldsymbol{Client}$ or $\widehat{\boldsymbol{Client}}$, respectively), in the limit of large component populations, approximating Equation 4.17 by Equation 4.9 and Equation 4.18 by Equation 4.13 is asymptotically correct. Informally, this is because the extra conditions of Equations 4.17 and 4.18 will have no local effect on the fluid component's state. Their only effect will be possibly to condition the state of the fluid component's previous cooperation partners. Then, if the maximum number of fluid

components which synchronise together in any cooperation stays constant whilst the number of fluid components is scaled up (as is the case for the GPEPA models considered in this thesis), it is reasonable that this error will vanish in the fluid limit. The convergence results of Section 6.4.2 will guarantee this formally.

In cases where there is more than one local source state for the passage time, we will see in Sections 5.5.1 and 5.5.2 that extra consideration is required to guarantee a correct result in the limit. The formal theoretical treatment of these issues is deferred till then and the proofs of the results in Section 6.4.2.

4.5 Conclusion

Passage-time measures in Markov chains are extremely useful for expressing probabilistic durations in real-world applications. Until now these calculations were limited to explicit state models and were limited by the size of system being analysed.

In this chapter, we have applied the results of Chapter 3 to allow the approximation of passage-time distributions. We have introduced the notion of global (Section 4.3) and individual (Section 4.4) passage times as being useful quantities in the context of massively-parallel systems and have then gone on to develop systematic methodologies for their approximation with ODE-based techniques.

We have shown how for large component populations, a global passage-time random variable can be approximated by an ODE-derived deterministic approximation (Section 4.3.1). Secondly, for global passage times, accurate upper and lower approximations of the cumulative distribution function can be obtained (Section 4.3.2).

For individual passage times, we have seen how the entire cumulative distribution can be accurately approximated. We have considered individual passage times both in the steady-state (Section 4.4.1) and in transient (Section 4.4.2) regimes.

Chapter 5

Unified stochastic probes

Key contributions
Immediate action GPEPA extension
Unified stochastic probe framework
Action-counting ODE approximations
Differential-equation analysis of probe-specified performance measures

5.1 Introduction

Performance modelling 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. In the previous chapter, we considered the latter problem and developed techniques for the ODE-based approximation of passage-time quantities. These can be applied when the state-space size makes the extraction of the measure intractable to traditional approaches. In this chapter, we develop the *unified stochastic probe* framework which can be used to specify passage times accurately in detailed performance models. We proceed to show how the analysis techniques of the last chapter can be extended to passage-time measures specified in this framework to yield an approach which is aimed at simultaneously addressing both the issues of measure specification and evaluation.

Furthermore, with many performance modelling formalisms, the feature or property to be measured, timed, counted or verified has to be captured explicitly in the state space of the underlying model to enable performance analysis to take place. Examples of this might 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 a modeller has 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 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. We will show that with unified stochastic probes, performance models can have complex queries constructed about them which do not rely on key features being present explicitly in the model.

Unified stochastic probes are specified with a regular expression syntax which is translated into a fragment of process algebra. This process algebra term is then composed with a part of the model which is to be measured and the resulting composed model can be analysed for the performance measure of interest.

Unified stochastic probes will also permit the specification of composed measurements formed by combining individual measurements captured from different parts of the model. The results of these local measurements are communicated between probes through the use of immediate signals. In order to support this functionality, we begin this chapter by developing an extension of the GPEPA process algebra to support the use of immediate actions and, furthermore, show how approximating systems of differential equations can still be obtained from such models.

We now give a summary of the key contributions of the work presented in this chapter which should also serve as a map of the chapter's structure.

Immediate action GPEPA extension In order to support immediate signalling between probe components, in Section 5.2 we develop the *immediate GPEPA (*i*GPEPA)* extension to GPEPA. This permits the use of immediate actions in GPEPA models. Our main contribution here is to show how they may be integrated into the GPEPA formalism in such a way as to ensure that the differential-equation approximation techniques of the previous two chapters remain available. Specifically, we achieve this by disallowing cooperation on immediate actions across component groups. Fortunately this does not prevent local probes monitoring different parts of fluid components from communicating with each other using immediate signals.

Unified stochastic probe framework We bring together significant existing stochastic probe mechanisms to create the unified stochastic probe framework in Section 5.3. Although, we base the framework around the GPEPA formalism, it could, in theory, be applied to any suitable behavioural performance modelling formalism. We base our approach on the original stochastic probe definition [8], which used a regular expression specification to define the start and end events of a measure. We also include the following useful extensions:

Immediate signalling permits measurements to be started and terminated precisely with immediate signals between probes using the immediate action functionality made available in the *i*GPEPA extension. This also allows the modeller to have many slave probes signalling to a master probe to permit more complex internal measurement features to be captured.

Flexible location-based specification allows unified stochastic probes to be placed precisely within a complex model architecture using a pattern rewriting syntax [80].

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 [79]. This gives the modeller the ability to capture measures that might also have been specified in tools such as DNAmaca [164], PRISM [171] and Möbius [90] or with formalisms such as CSL [11], asCSL [15] and CSL^{TA} [95].

Action-counting ODE approximations In addition to component-count approximations for *i*GPEPA models, in Section 5.4, we also introduce similar systems of differential equations approximating moments of action-counting processes. In Section 5.6, we will

see that these are particularly useful for evaluating probe-specified global passage-time queries, which depend on the observation of action behaviour from the model.

Differential-equation analysis of probe-specified performance measures We show how the differential-equation approximation techniques of the previous chapter can be extended to perform efficient analysis of probe-specified performance queries in Sections 5.5 and 5.6 and demonstrate this in Section 5.7 on an example model of a massively-distributed wireless network.

5.2 Immediate GPEPA

In Chapter 3, we introduced grouped PEPA or GPEPA, a simple extension of the stochastic process algebra PEPA, useful for applying ODE-based techniques to massively-parallel models. In this section, we introduce *immediate GPEPA (*i*GPEPA)* which adds immediate action functionality to GPEPA models. This is useful in itself but our key motivation here is to provide a framework in which stochastic probes [8] can be used to specify performance measures which can then be analysed using the ODE-based techniques of the previous chapters. Specifically, immediate actions are very useful to allow communication between probes [9], for example many slave probes can signal to a master probe to permit complex measurements to be captured. Immediate signalling between probes has already been considered by Argent-Katwala et al. [9]. However, this is implemented on top of the semi-Markov PEPA [49] extension to PEPA which supports generally-distributed state holding times and thus immediate actions as a special case. On the other hand, our approach explicitly considers the removal of *vanishing states* at the level of fluid components (Section 5.2.1) which is necessary if ODE techniques are to be applied.

Support for immediate actions in a Markovian modelling formalism is not in itself a new idea. Generalised stochastic Petri nets (GSPNs) [5] introduced immediate transitions into the stochastic Petri net (SPN) [20] formalism and in terms of process algebras, the languages EMPA [31], MoDeST [83], SPADES [128], TIPP [144, 209] and SM-PEPA [49] all have some level of support for immediate actions. As above, though, our key contribution here is to introduce immediate actions in such a way as to ensure that the ODE-analysis techniques of the earlier chapters are still applicable. In particular, we will restrict cooperation on immediate actions to be allowed only within fluid components and not across component groups. This restriction notwithstanding, we will see that the resulting language has more than enough expressiveness to allow ODE analysis of probe-specified performance measures.

We will begin by defining an extension of the original process algebra PEPA to include immediate actions. For consistency, we will call this *immediate PEPA*, or i*PEPA* . In GPEPA, we recall that fluid components were allowed to be any standard *PEPA* component and, naturally, in *i*GPEPA, they will be i*PEPA* components.

5.2.1 Immediate PEPA

An *immediate PEPA (*i*PEPA)* component is simply defined to be any standard PEPA process algebra component with the additional possibility of immediate actions. Syntactically, an

iPEPA component is specified by the standard PEPA grammar (see Equation 2.4), augmented with immediate actions:

$$\begin{aligned} S &::= (\alpha, r).S \;\mid\; [\mathsf{a}, w].S \;\mid\; S + S \;\mid\; \boldsymbol{C}_S \\ P &::= P \underset{L}{\bowtie} P \;\mid\; P/L \;\mid\; S \;\mid\; \boldsymbol{C}_P \end{aligned} \tag{5.1}$$

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

The new option in the above grammar, *immediate prefix*, $[\mathsf{a}, w].S$ is intended to behave like the component $(\alpha, r).S$, except that 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 will be resolved probabilistically using the weight parameter. We also allow $\mathsf{a}.S$ as shorthand for $[\mathsf{a}, 1].S$ in situations for which the modeller deems a weighting irrelevant (for example, if the modeller knows by construction either that this immediate prefix will never be concurrently enabled with another; or that it might, but not in a way which will affect the timed behaviour of the composed model).

Now that immediate as well as timed actions are allowed it is important to understand how they may interact. This has been considered by Argent-Katwala et al. [9]. We have adopted some of those design decisions here. In particular, we have enforced the separation of timed and immediate activities by drawing their types from two disjoint sets $\mathcal{A}_t$ and $\mathcal{A}_i$ respectively. This prevents direct cooperation between timed and immediate actions.

A semantics for immediate actions in PEPA is discussed in detail by Argent-Katwala et al. [9], and given implicitly as a special case of semi-Markov PEPA [49]. For clarity, we instead present an explicit structured operational semantics for iPEPA, which combines both timed actions (represented using a solid arrow) and immediate actions (represented using a dashed arrow) in Figure 5.1. It requires two auxiliary definitions: the *total weight* function, which computes the sum of the weights of all concurrently-enabled immediate activities for a given iPEPA component; and 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.

Definition 5.2.1 (iPEPA:: Total weight)**.** *Let P be an* i*PEPA component. Then the total weight of concurrently-enabled immediate activities is $w(P)$, defined very straightforwardly as follows.*

$$\begin{aligned} w((\beta, \lambda).P) &:= 0 \\ w([\mathsf{a}, w].P) &:= w \\ w(P + Q) &:= w(P) + w(Q) \\ w(P/L) &:= w(P) \\ w(P \underset{L}{\bowtie} Q) &:= w(P) + w(Q) \end{aligned}$$

Definition 5.2.2 (iPEPA:: Total probability)**.** *Let P be an* i*PEPA component and $\mathsf{a} \in \mathcal{A}_i$ an immediate action type. Then the total probability of concurrently-enabled immediate a-activities is $z_{\mathsf{a}}(P) := \sum_{P \xrightarrow{(\mathsf{a}, \cdot, z)}} z$.*

Prefix

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

Competitive Choice

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

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

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

Cooperation

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

$$\frac{P \overset{(\mathsf{a},\cdot,z)}{\dashrightarrow} P'}{P \underset{L}{\bowtie} Q \overset{(\mathsf{a},\cdot,Z_1)}{\dashrightarrow} P' \underset{L}{\bowtie} Q}\ \mathsf{a} \in \mathcal{A}_i, \mathsf{a} \notin L \qquad \frac{Q \overset{(\mathsf{b},\cdot,z)}{\dashrightarrow} Q'}{P \underset{L}{\bowtie} Q \overset{(\mathsf{b},\cdot,Z_2)}{\dashrightarrow} P \underset{L}{\bowtie} Q'}\ \mathsf{b} \in \mathcal{A}_i, \mathsf{b} \notin L$$

$$\frac{P \xrightarrow{(\alpha,\lambda)} P' \quad Q \xrightarrow{(\alpha,\mu)} Q'}{P \underset{L}{\bowtie} Q \xrightarrow{(\alpha,R)} P' \underset{L}{\bowtie} Q'}\ \alpha \in \mathcal{A}_t, \alpha \in L \qquad \frac{P \overset{(\mathsf{a},\cdot,z_1)}{\dashrightarrow} P' \quad Q \overset{(\mathsf{a},\cdot,z_2)}{\dashrightarrow} Q'}{P \underset{L}{\bowtie} Q \overset{(\mathsf{a},\cdot,Z)}{\dashrightarrow} P' \underset{L}{\bowtie} Q'}\ \mathsf{a} \in \mathcal{A}_i, \mathsf{a} \in L$$

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 $Z := \frac{w(P)}{w(P \underset{L}{\bowtie} Q)} z_1 \frac{z_2}{z_{\mathsf{a}}(Q)} + \frac{w(Q)}{w(P \underset{L}{\bowtie} Q)} z_2 \frac{z_1}{z_{\mathsf{a}}(P)}$

Hiding

$$\frac{P \xrightarrow{(\alpha,\lambda)} P'}{P \backslash L \xrightarrow{(\tau,\lambda)} P'}\ \alpha \in \mathcal{A}_t, \alpha \in L \qquad \frac{P \xrightarrow{(\alpha,\lambda)} P'}{P \backslash L \xrightarrow{(\alpha,\lambda)} P'}\ \alpha \in \mathcal{A}_t, \alpha \notin L$$

$$\frac{P \overset{(\mathsf{a},\cdot,z)}{\dashrightarrow} P'}{P \backslash L \overset{(\tau,\cdot,z)}{\dashrightarrow} P'}\ \mathsf{a} \in \mathcal{A}_i, \mathsf{a} \in L \qquad \frac{P \overset{(\mathsf{a},\cdot,z)}{\dashrightarrow} P'}{P \backslash L \overset{(\mathsf{a},\cdot,z)}{\dashrightarrow} P'}\ \mathsf{a} \in \mathcal{A}_i, \mathsf{a} \notin L$$

Constant

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

Figure 5.1: Operational semantics for immediate PEPA.

The operational semantics determines a transition relation on *i*PEPA components, consisting both of immediate and timed transitions. For an *i*PEPA component P, we define its set of *derivative states* $ds(P)$ to be all *i*PEPA components reachable from it by either timed or immediate transitions according to the operational semantics. Formally: $ds(P)$ is the smallest set of *i*PEPA components such that $P \in ds(P)$ and if for any $P_1 \in ds(P)$, $P_1 \xrightarrow{(\cdot,\cdot)} P_2$ or $P_1 -\!\!\stackrel{(\cdot,\cdot,\cdot)}{\rightarrow} P_2$; then $P_2 \in ds(P)$. Immediate transitions have three parameters. 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 Figure 5.1 for model components. Finally, the third parameter is a probability value determining a probability distribution over all concurrently-enabled immediate transitions. This distribution will be 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.

At the level of sequential components, these probabilities are computed directly using the weights supplied by the modeller. At the level of model components, say $P \underset{L}{\bowtie} Q$, that is, where multiply-enabled immediate transitions are introduced by concurrency rather than choice, we adopt a different approach to the computation of the probabilities. Specifically, we first use the weights to choose probabilistically a side of the cooperation to be allowed to proceed with an immediate transition; for example, P is allowed to proceed with probability $\frac{w(P)}{w(P)+w(Q)}$. Assuming P proceeds, then if P also enables concurrent immediate transitions itself, the probability distribution already computed on its transitions is used to determine the one to be performed. If the immediate action type, say $\mathsf{a} \in \mathcal{A}_i$, of the chosen transition is shared, that is, in the cooperation set L, we must choose a corresponding transition of that action type for Q to perform. This is achieved by normalising the distribution on a-transitions of Q, that is, using the distribution defined by $\frac{z}{z_{\mathsf{a}}(Q)}$ for each transition $Q -\!\!\stackrel{(\mathsf{a},\cdot,z)}{\rightarrow}$.

Vanishing state removal

Assuming that an *i*PEPA component satisfies two natural regularity conditions, its transition system has a straightforward translation to a continuous-time Markov chain (CTMC), as in the case of PEPA without immediate actions. When an *i*PEPA 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 *i*PEPA component's transition system;
- *deterministic initial behaviour*, meaning that there may be at most one path of immediate transitions emanating from a well-behaved *i*PEPA component's initial state.

We will require that all fluid components in *i*GPEPA models are well behaved. Formal definition of these conditions is given in Appendix C.1.1.

If an *i*PEPA 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 (CTMC), but the state space of a fluid component P is not $ds(P)$, rather it is the set of *non-vanishing derivative states* $ds^*(P)$. This is the set of derivative states of P which do not enable any immediate actions, i.e. $ds^*(P) := \{P' \in ds(P) \ : \ P' \not\xrightarrow{}\}$.

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

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

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 $\mathcal{I} := (\mathsf{a}_1, \ldots, \mathsf{a}_K)$. The extra *sequence* parameter $\mathcal{I}$ extends the transition system so that it is still possible to determine in the derived transition system when immediate actions are performed and in which order.[1] The derived transition system will be used most prominently in Section 5.4 when we construct action-counting differential equations for iGPEPA models.

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 non-vanishing derivative state reachable by following immediate transitions from P. This non-vanishing derivative state is taken to be the initial state of the iPEPA component in the derived transition system.

The derivation graph determined by the derived transition system can then be interpreted naturally as a continuous-time Markov chain, whose states are given by the non-vanishing derivative states and each arc represents a transition at the rate labelling the arc. This is the *underlying continuous-time Markov chain* of an iPEPA model.

5.2.2 Constructing iGPEPA models from iPEPA fluid components

An iGPEPA model is simply a GPEPA model as defined in Section 3.2 except that fluid components can be any well-behaved *immediate* PEPA model (as opposed to just a standard PEPA model). Note that in the syntactic definition of a GPEPA model (Equation 3.3), synchronisation between component groups may only happen on timed action types, i.e. on action types in $\mathcal{A}_t$. This has important ramifications in the case of iGPEPA, disallowing cooperation between component groups on immediate actions. We will see that this restriction allows meaningful extension of the ODE-based analysis techniques to iGPEPA.

As in the case of GPEPA, we wish to define the operational semantics for iGPEPA to be identical to those obtained by flattening the model, that is, by removing the group labels and replacing $\wr$ combinators with $\|$. This can be achieved for iGPEPA by using the same flattening function ($\mathcal{F}(\cdot)$) as defined for GPEPA in Definition 3.2.1.

[1] 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.

Then analogously to the case of GPEPA, for any two iGPEPA models, G_1 and G_2, we say $G_1 \xrightarrow{(\alpha,r)} G_2$ or $G_1 \xrightarrow{(\mathsf{a},w,z)} G_2$ if and only if $\mathcal{F}(G_1) \xrightarrow{(\alpha,r)} \mathcal{F}(G_2)$ or $\mathcal{F}(G_1) \xrightarrow{(\mathsf{a},w,z)} \mathcal{F}(G_2)$, respectively, counting also the number of such transitions. We may thus extend the notion of non-vanishing derivative states to iGPEPA models. Removing the vanishing states of an iGPEPA model gives rise to its derived transition system and thus its underlying CTMC. The removal of vanishing states at the level of component groups is however impractical since to do so naïvely would require the expansion of the state space of the entire model. Fortunately, this is avoidable due to the fact that there is no cooperation between component groups on immediate actions.

Indeed, it is easy to see that due to this fact, any non-vanishing derivative state of an iGPEPA model must have all constituent fluid components in non-vanishing iPEPA states. That is to say that since there is no cooperation on immediate actions across component groups, no vanishing state of a fluid component can become non-vanishing under composition as part of a grouped model and vice-versa. Therefore removing vanishing states from the constituent fluid components *before* composing them to form a grouped model results in the same derived transition system as would be obtained by removing vanishing states at the level of component groups. This is useful because fluid components can be expected to have manageable state spaces and therefore this approach represents a tractable way of characterising the derived transition system of an iGPEPA model.

5.3 Unified stochastic probes

Stochastic probes are a query mechanism for stochastic process algebras originally introduced by Argent-Katwala et al. [8], extended with location specifications in Argent-Katwala et al. [9], state awareness by Clark and Gilmore [79] and the introduction of a probe-placement language by Clark and Gilmore [80]. They are used to specify events which determine the start and end of a desired passage-time measure.[2] In particular, we shall see how we may use stochastic probes to specify elegantly the types of passage-time measurements discussed in the previous chapter. One particular benefit of stochastic probes is that they allow the specification of measures depending on derived behaviour not expressed directly in the state space of the original model. We will see that stochastic probes therefore dispense with the need to manually modify the state space of components in order to extract passage-time measurements, as was necessary in the last chapter. For complicated measurements of detailed systems, this eliminates a potentially large source of modelling error.

Unified stochastic probes are formalism independent and use a regular expression syntax, based on the action label set of the model, combined with an optional state-based predicate to define the probe itself (similar to, for example, the asCSL [15] or CSL$^{\text{TA}}$ [95] automaton). Traditionally, a stochastic probe is translated into a fragment of stochastic process algebra that can be composed with a stochastic process algebra model, simultaneously facilitating both separation and reuse of the model and the measurement specification. The new derived model can then be analysed for the performance measure of interest.

As an example, the following is a probe that could be attached to a process model that emits

[2] They can also be used to specify other steady-state and transient measures [8], but we focus on the passage-time measure in this chapter.

initialise, *fetch* and *fail* actions:

$$\mathit{initialise} : \mathsf{start}, (\mathit{fetch}[3] \backslash \mathit{fail}) : \mathsf{stop}$$

This would specify a probe that would start a measurement after seeing an *initialise* action and terminate it on seeing three consecutive *fetch* actions without seeing any *fail* actions. The start and stop labels are reserved signals specifying the beginning and end of the measurement.

Unlike asCSL or CSL$^{\text{TA}}$, 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 model component that the modeller is interested in measuring. Unified stochastic probes also permit many distinct *local* probes to be created and distributed over the model structure. The results of local probes can in turn be observed via immediate signals by a global probe. The global probe aggregates the local probe observations to allow the definition of a more sophisticated passage-time measure. This would, for instance, allow us to define a passage time that terminates on observing only 50 clients reaching a certain state in a client–server system with many hundreds of clients.

Finally, we note that CSL$^{\text{TA}}$ supports multiple time constraints. This allows, for example, the specification of a measure where, with some probability, the first part of an event sequence is constrained to complete within one given time bound, and the second part within some other given time bound. As it stands, this is not directly possible within the unified stochastic probe framework, although, it would be straightforward to augment the probe language to allow support for the specification of such measurements. One way of doing this would be to introduce multiple indexed start and stop label pairs to start and stop multiple clocks, each associated with a different time constraint.

However, particularly when nested time intervals can be introduced in this manner, the corresponding analysis becomes more involved (even when only traditional, explicit-state approaches are employed). Indeed, in the case of CSL$^{\text{TA}}$, Donatelli et al. [95] map the product model–automaton space onto a *Markov regenerative process (MRP)* for which explicit-state solution techniques do exist [109]. However, this approach only works where there is at most one concurrently-enabled clock. This additional complexity in the analysis, before ODE-based techniques are even considered, is why we do not consider such an extension of the probe language in this thesis.

5.3.1 Probe syntax

In this thesis, we will use stochastic probes inspired by the definitions of Clark and Gilmore [79] augmented with the unique *i*GPEPA component group labels. These will provide a convenient mechanism by which to identify specific components and to locate measurement probes. We will build on and extend the notions of *local* and *global* probes as discussed by Clark and Gilmore [79] to facilitate the application of ODE-based analysis techniques later in Sections 5.5 and 5.6.

For our purposes, *local* probes are probes which can be attached to observe fluid components or their sub-components and a *global* probe is a probe which observes the entire *i*GPEPA 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 5.3.3, we will introduce location

specifications for local probes which will define exactly which fluid component they observe, that is, where they are placed within a model. Every probed model must include exactly one global probe but may utilise zero or many local probes.

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:

$$Probe_l ::= R_l^s \mid R_l^{s\hookleftarrow} \tag{5.2}$$

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

$$\begin{array}{rll} R_l^s ::= & R_l^s,\, R_l^s & \text{sequence} \\ \mid & R_l : signal & \text{transmitted signal} \end{array} \tag{5.3}$$

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 expressive power of an extended regular-expression-based language, suited to passage-time specification:

$$\begin{array}{rll} R_l ::= & R_l,\, R_l & \text{sequence} \\ \mid & R_l \mid R_l & \text{choice} \\ \mid & R_l;\, R_l & \text{both} \\ \mid & R_l[n] & \text{iterate } n \text{ times} \\ \mid & R_l[m,n] & \text{iterate } m \text{ to } n \text{ times} \\ \mid & R_l^? & \text{zero or one} \\ \mid & R_l^+ & \text{one or more} \\ \mid & R_l^* & \text{zero or more} \\ \mid & R_l / R_l & \text{reset} \\ \mid & R_l \varnothing R_l & \text{fail} \\ \mid & R_l^! & \text{not} \\ \mid & . & \text{any action or signal} \\ \mid & action & \text{eventual specific action or signal} \\ \mid & \overline{action} & \text{subsequent specific action or signal} \\ \mid & \epsilon & \text{empty action or signal sequence} \end{array} \tag{5.4}$$

where $action \in \mathcal{A} \setminus \{\tau\}$ is an action (or signal) type. For the purposes of matching regular expressions against traces of observed actions, transitions of the hidden action type τ are ignored.

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 Equation 5.3). For this reason, we will 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 (Equation 5.3) independently of the previous one.

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

We provide the shorthand for eventual observations since this is very often how passage-time queries are specified. The modeller 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 $\overline{fetch}$: end will only transmit the end signal if the *first* action it observes is *fetch*. Otherwise, it will *never* match and thus never transmit the end signal. On the other hand, the local probe specified by *fetch* : end will match and transmit 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 non-standard 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 $R_l \oslash R_l$ 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 5.3.2 by means of the translation from probes to *i*PEPA components.

Global probes

A global probe may observe actions and signals like a local probe, but can also directly query the state of the *i*GPEPA 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:

$$\begin{aligned} Probe_g ::= {} & R_g : \mathsf{start},\ R_g : \mathsf{stop} \\ \mid {} & R_g : \mathsf{start},\ R_g : \mathsf{stop}^{\hookleftarrow} \end{aligned} \tag{5.5}$$

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

$$R_g ::= \{pred\}R_g \mid R_g, R_g \mid \ldots \mid action \mid \overline{action} \mid \epsilon \tag{5.6}$$

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 non-repeating case, for a given trace of the probed model, both the start and stop signals will only ever be transmitted at most once. The passage time being measured is just the difference between the time instants at which these two signals were transmitted. Transient passage-times such as the transient individual passage-times discussed in the previous chapter and global passage times would be specified using a global probe of this kind.

On the other hand, repeating global probes will 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. Therefore, this is the type of global probe involved in the specification of the class of steady-state individual passage times introduced in the previous chapter.

Finally, the state guard predicate, *pred*, is given by:

$$\begin{aligned} pred ::= {} & \mathsf{true} \mid \mathsf{false} && \text{boolean} \\ \mid {} & \neg pred && \text{not} \\ \mid {} & pred \vee pred && \text{disjunction} \\ \mid {} & b_expr && \text{expression} \\ b_expr ::= {} & r_expr \succeq r_expr && \text{comparison} \\ r_expr ::= {} & H : P && \text{component count in group} \\ \mid {} & int && \text{number} \\ \mid {} & r_expr \oplus r_expr && \text{arithmetic} \\ \succeq ::= {} & = \mid \geq \mid \leq && \text{relational ops} \\ \oplus ::= {} & + \mid - && \text{binary ops} \end{aligned}$$

where, if G is the iGPEPA model to which the global probe will 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:

$$\begin{aligned} & s \models \mathsf{true} && \text{for all } s \\ & s \models \mathsf{false} && \text{for no } s \\ & s \models \neg\psi && \text{iff } s \not\models \psi \\ & s \models \psi_1 \vee \psi_2 && \text{iff } s \models \psi_1 \vee s \models \psi_2 \\ & s \models b_expr && eval(b_expr, s) \end{aligned}$$

where b_expr is, as above, a boolean function of $H : P$ expressions, which reference a specific fluid component P in a component group H. The *eval* function evaluates the boolean function by dereferencing the $H : P$ expressions as the number of P fluid 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 behaviour specified by R_g subject to the state-based guard $\{pred\}$ being true when R_g starts to be matched.

5.3.2 Translating a local probe to *i*PEPA

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 *i*PEPA component. The resulting component can then be attached at various points within a 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 process algebra components and composed with the original model as required. The resulting composite model can then be analysed as a standalone model using traditional techniques.

Where fluid analysis techniques are to be used, we translate local probes directly into *i*PEPA components and compose them with the model at the specified locations. However, global probes are not translated into process algebra components. The translation of global probes directly into process algebra components and their subsequent composition with the model of interest would in fact technically be prohibited by our restriction that *i*GPEPA 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 synchronisation on immediate actions at a global level leads to ill-defined systems of differential equations similar to those discussed in Section 3.2.1. Instead, global probes are interpreted directly depending on the type of performance measure being computed as will be described in Sections 5.5 and 5.6.

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

The translation of local probes to *i*PEPA components will follow the standard approach of first converting the regular expression to a *deterministic finite automaton (DFA)* [e.g. 227]. A DFA is a labelled 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 *i*PEPA component which can be synchronised with the observed fluid component. We begin by showing how a regular expression R_l given by the grammar of Equation 5.4 may be translated to a DFA.

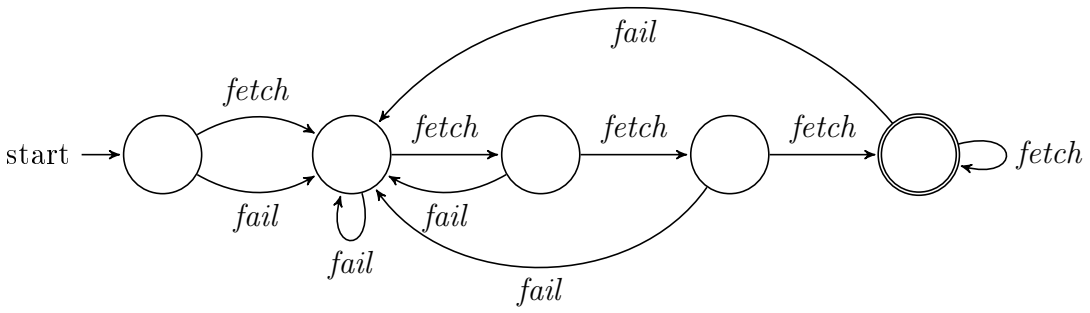
$$\begin{aligned}
\boldsymbol{Pb} &\stackrel{def}{=} (fetch, \top).\boldsymbol{Pb}_1 + (fail, \top).\boldsymbol{Pb}_1 \\
\boldsymbol{Pb}_1 &\stackrel{def}{=} (fail, \top).\boldsymbol{Pb}_1 + (fetch, \top).\boldsymbol{Pb}_2 \\
\boldsymbol{Pb}_2 &\stackrel{def}{=} (fail, \top).\boldsymbol{Pb}_1 + (fetch,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

Figure 5.2: Translation of the local probe $(fetch \mid fail), (fetch[3] \backslash fail)$ to an *i*PEPA component with $\mathcal{A}(P) = \{fetch, fail\}$.

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. 227]. 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, $R_l^1 \varnothing R_l^2$ 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.2. By way of example, the local probe regular expression:

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

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



At this stage, DFA minimisation algorithms [e.g. 227] could be applied to reduce its size if desired. The next stage is the translation of the DFA to an *i*PEPA component, which is straightforward. Each state of the DFA maps to a derivative state of the *i*PEPA component. Each arc with a timed action is translated to a passive enabling of that action type in the component, and each arc with an immediate action is translated to an immediate enabling of that action type. For the example above, the resulting *i*PEPA component is given in Figure 5.2.

A complete local probe may also transmit signals and possibly be repeating according to the grammars of Equations 5.2 and 5.3. Indeed, a general instance of the grammar of Equation 5.3 has the form $R_l^1 : \mathsf{s}_1, \ldots, R_l^k : \mathsf{s}_k$. We may construct *i*PEPA 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

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

Figure 5.3: Translation of the local probe $(fetch \mid fail) : \mathsf{begin}, (fetch[3] \backslash fail) : \mathsf{end}$ to an iPEPA component with $\mathcal{A}(P) = \{fetch, fail\}$.

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 will cause the probe component to emit an s_i signal before beginning to match the next regular expression in the sequence.

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 non-repeating case, this immediate action instead ends in a new derivative state which loops on all actions in $\mathcal{A}(P) \setminus \{\tau\}$ (so as not to block the observed component under synchronisation).

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

$$(fetch \mid fail) : \mathsf{begin}, (fetch[3] \backslash fail) : \mathsf{end}$$

The complete iPEPA component corresponding to this local probe is given in Figure 5.3.

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 fluid component $P \underset{\mathcal{A}(P)\setminus\{\tau\}}{\bowtie} \boldsymbol{Pb}$. If the fluid component P happens to enable actions which might not be observed by the probe, say, for the example above that $\mathcal{A}(P) = \{fetch, fail, initialise\}$ instead, then the translation of the probe may involve a large number of self-loop transitions. See Figure 5.4 for the iPEPA component which would result in this case. 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. So *even if* $\mathcal{A}(P) = \{fetch, fail, initialise\}$, as long as we compose the fluid component and the probe as $P \underset{\{fetch, fail\}}{\bowtie} \boldsymbol{Pb}$ instead of $P \underset{\{fetch, fail, initialise\}}{\bowtie} \boldsymbol{Pb}$, then we may use the more succinct probe definition of Figure 5.3 instead of Figure 5.4.

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

Figure 5.4: Translation of the local probe $(fetch \mid fail) : \mathsf{begin}, (fetch[3] \backslash fail) : \mathsf{end}$ to an iPEPA component with $\mathcal{A}(P) = \{fetch, fail, initialise\}$.

So in future, we will 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 as above. 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 fluid 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 $\overline{fetch} : \mathsf{begin}$. If P does not immediately perform a *fetch* action, the regular expression will never match. 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 $R_l^1 : \mathsf{s}_1, \ldots, R_l^k : \mathsf{s}_k^{\hookleftarrow}$. 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 been accepted, that is, to reach some $S \underset{*}{\bowtie} Pb^i_{\mathrm{Acc}}$ 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 a fluid 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 behaviour.

As an aside, we declare that $P \underset{*}{\bowtie} Probe_l$ shall also have the same meaning if $Probe_l$ is a regular expression definition of a local probe according to Equation 5.2, as opposed to the iPEPA component obtained by the translation of the regular expression Pb. Finally, we note that it would be straightforward to automate the translation process described above starting from

the initial regular expression specification of the local probe. Therefore, with the right tool support, the modeller need not interact with the evaluation process further after specifying the regular expression defining the measurement.

5.3.3 Locating a local probe

To locate a local probe, we use a modified version of the model transformation language presented by Clark and Gilmore [80], extended to use the component group labels for the purpose of uniquely locating the probe. Thus in the client–server model $\boldsymbol{CS}(n, m)$ of Section 3.2.3, we might want to position a local probe, $\boldsymbol{Pb}$ around one of the $\boldsymbol{Client}$ components in the **Clients** group. To do this we would deploy a model transformation:

$$\mathbf{Clients}\{\boldsymbol{Client}[?n]\} \Longrightarrow \mathbf{Clients}\{(\boldsymbol{Client} \underset{*}{\bowtie} \boldsymbol{Pb}) \wr\wr \boldsymbol{Client}[?n-1]\}$$

This is used to pattern match against the definition of $\boldsymbol{CS}(n, m)$ 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 **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 $\boldsymbol{Client}$ is specified. It permits subsequent arithmetic operation to be performed on $?n$ in the transformed model. We might also wish to abstract away from the specific fluid components which make up the **Clients** group, in which case, we could use the transformation:

$$\mathbf{Clients}\{?p[?n]\} \Longrightarrow \mathbf{Clients}\{(?p \underset{*}{\bowtie} \boldsymbol{Pb}) \wr\wr ?p[?n-1]\}$$

which matches against a homogeneous array of any type of fluid component in a group labelled by **Clients**. Formally we draw the syntax of a location specification term on the model G 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	fluid component
|	$cpt \wr\wr cpt$	fluid component parallelism
|	$cpt[expr]$	fluid component array
cpt ::=	P	*i*PEPA component
|	$cpt \underset{actions}{\bowtie} cpt$	*i*PEPA cooperation
|	$?p$	*i*PEPA 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 $H' \in \mathcal{G}(G)$, $L \subseteq \mathcal{A}_t \setminus \{\tau\}$, $int \in \mathbb{Z}_+$ and n and p are drawn from a set of variable names.

As a final example of probe placement, consider a situation in which ***Server*** components operate in pairs, synchronising on some set R of actions, perhaps represented by the following iGPEPA model:

$$\mathbf{Clients}\{\boldsymbol{Client}[n]\} \underset{L}{\bowtie} \mathbf{Servers}\{(\boldsymbol{Server} \underset{R}{\bowtie} \boldsymbol{Server})[m]\}$$

Perhaps we are interested in monitoring the behaviour of one such pair. To do this, we would 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:

$$\begin{aligned}&\mathbf{Servers}\{(\boldsymbol{Server} \underset{R}{\bowtie} \boldsymbol{Server})[?n]\} \Longrightarrow \\ &\quad \mathbf{Servers}\{((((\boldsymbol{Server} \underset{*}{\bowtie} \boldsymbol{Pb}_1) \underset{R}{\bowtie} (\boldsymbol{Server} \underset{*}{\bowtie} \boldsymbol{Pb}_2)) \underset{*}{\bowtie} \boldsymbol{Pb}_m) \wr\wr (\boldsymbol{Server} \underset{R}{\bowtie} \boldsymbol{Server})[?n-1]\}\end{aligned} \tag{5.7}$$

Nested local probes are automatically supported since a fluid iPEPA component becomes another fluid iPEPA component under probe composition.

It is important to realise 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 in order to access some derived performance quantity such as a passage-time measure.

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:

$$\begin{aligned}\mathit{ProbedModel} ::= {} & \mathit{Probe}_g \ \mathsf{observes}\ \{\mathit{Probe}_l\} \\ & \mathsf{where}\ \{\mathit{Location}\} \\ & \mathsf{in}\ \ G \\ \mid {} & \mathit{Probe}_g \ \mathsf{in}\ G\end{aligned} \tag{5.8}$$

where $\{\mathit{Probe}_l\}$ is a list of local probes and $\{\mathit{Location}\}$ is a list of location specifications used to place the local probes within the iGPEPA model G.[3] We assume that the location specification transformations are applied in the order of enumeration of this list.

5.3.4 Example measure specifications

In this section, we provide a few example passage-time measurements, specified in the language of unified stochastic probes. Then in Sections 5.5 and 5.6, we will show how they may be analysed using extensions of the ODE-based techniques presented in the previous chapter.

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

We consider again the client–server GPEPA model $\boldsymbol{CS}(n, m)$ from Section 3.2.3. A global passage-time query (as introduced in Section 4.3 of the last chapter) using state-based specification on this model might ask *how long until n of the servers are simultaneously in state* $\mathbf{Server_{broken}}$. This could be captured by the probed model specification:

$$\begin{aligned}\boldsymbol{PM}_1 \stackrel{def}{=} \epsilon : \mathsf{start}, \{\mathbf{Servers} : \boldsymbol{Server_{broken}} \geq n\}\epsilon : \mathsf{stop}\\ \mathsf{in} \quad \boldsymbol{CS}(n, m)\end{aligned}$$

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

$$\begin{aligned}\boldsymbol{PM}_2 \stackrel{def}{=} \epsilon : \mathsf{start}, \mathit{break}[n] : \mathsf{stop}\\ \mathsf{in} \quad \boldsymbol{CS}(n, m)\end{aligned}$$

No local probes or location specifications are required in either of the above cases. If, however, we are instead interested in the query *how long before n* different *servers each perform a break-action*, we would need to employ a local probe which would be attached to each ***Server*** component:

$$\begin{aligned}\boldsymbol{PM}_3 \stackrel{def}{=} \epsilon : \mathsf{start}, \mathsf{end}[n] : \mathsf{stop}\\ \mathsf{observes} \quad \boldsymbol{Probe}_l \stackrel{def}{=} \mathit{break} : \mathsf{end}\\ \mathsf{where} \quad \mathbf{Servers}\{\boldsymbol{Server}[?n]\} \Longrightarrow \mathbf{Servers}\{(\boldsymbol{Server} \underset{*}{\bowtie} \boldsymbol{Probe}_l)[?n]\}\\ \mathsf{in} \quad \boldsymbol{CS}(n, m)\end{aligned}$$

In this case, the local probe is necessary to extract the derived behaviour from each ***Server*** component of having *already completed a break-action*. After observing this it signals to the global probe using the end signal.

Transferring our attention to individual passage-time queries (as introduced in Section 4.4), we might simply be interested in *how long it takes a particular client to complete its first think-action*. In order to specify this measurement we may again use a local probe like that above to capture the derived behaviour of having completed a *think*-action, but we apply it to only one ***Client*** component:

$$\begin{aligned}\boldsymbol{PM}_4 \stackrel{def}{=} \mathsf{begin} : \mathsf{start}, \mathsf{end} : \mathsf{stop}\\ \mathsf{observes} \quad \boldsymbol{Probe}_l \stackrel{def}{=} \epsilon : \mathsf{begin}, \mathit{think} : \mathsf{end}\\ \mathsf{where} \quad \mathbf{Clients}\{\boldsymbol{Client}[?n]\} \Longrightarrow \mathbf{Clients}\{(\boldsymbol{Client} \underset{*}{\bowtie} \boldsymbol{Probe}_l) \wr\wr \boldsymbol{Client}[?n-1]\}\\ \mathsf{in} \quad \boldsymbol{CS}(n, m)\end{aligned}$$

Note that it is *not* true that the same measurement could be specified with just the global probe $\epsilon : \mathsf{start}, \mathit{think} : \mathsf{stop}$ since this would terminate on seeing *any think*-action from *any* client. 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 data transfers without breaking*, we would need a more involved local probe, which again ensures that both of the *data*-actions and any *break*-actions are captured only from a single ***Server***

component:

$$\begin{aligned}
&\boldsymbol{PM}_5 \stackrel{def}{=} \mathsf{begin} : \mathsf{start}, \mathsf{end} : \mathsf{stop} \\
&\qquad \mathsf{observes} \quad \boldsymbol{Probe}_l \stackrel{def}{=} \epsilon : \mathsf{begin}, (data[2]) \backslash break : \mathsf{end} \\
&\qquad \mathsf{where} \quad \mathbf{Servers}\{\boldsymbol{Server}[?n]\} \Longrightarrow \mathbf{Servers}\{(\boldsymbol{Server} \underset{*}{\bowtie} \boldsymbol{Probe}_l) \wr\wr \boldsymbol{Server}[?n-1]\} \\
&\qquad \mathsf{in} \quad \boldsymbol{CS}(n,m)
\end{aligned}$$

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

$$\begin{aligned}
&\boldsymbol{PM}_6 \stackrel{def}{=} \mathsf{begin} : \mathsf{start}, \mathsf{end} : \mathsf{stop}^{\hookleftarrow} \\
&\qquad \mathsf{observes} \quad \boldsymbol{Probe}_l \stackrel{def}{=} reset : \mathsf{begin}, (data[2]) \backslash break : \mathsf{end}^{\hookleftarrow} \\
&\qquad \mathsf{where} \quad \mathbf{Servers}\{\boldsymbol{Server}[?n]\} \Longrightarrow \mathbf{Servers}\{(\boldsymbol{Server} \underset{*}{\bowtie} \boldsymbol{Probe}_l) \wr\wr \boldsymbol{Server}[?n-1]\} \\
&\qquad \mathsf{in} \quad \boldsymbol{CS}(n,m)
\end{aligned}$$

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

In the next section, we show how differential equations can be constructed from *i*GPEPA models in a similar fashion to the case of GPEPA. This prepares us for Sections 5.5 and 5.6 where we investigate the ODE-based analysis of individual and global passage-time queries specified using unified stochastic probes. We will then use these techniques to analyse both classes of passage-time measure as applied to a worked example of a massively-distributed wireless network in Section 5.7.

5.4 Differential equations from *i*GPEPA models

In this section, we will show how the approximating systems of ODEs given in Chapter 3 for GPEPA models can be naturally defined also for *i*GPEPA models. Recall from Section 5.2.2 that vanishing states may be removed at the fluid component level before their composition into a grouped model — this results in the same derived transition system and thus underlying CTMC. This means that the state of the aggregated CTMC of an *i*GPEPA model may be specified by the component counts of *non-vanishing* *i*PEPA fluid component derivative states. Furthermore, the aggregate rate of *i*GPEPA transitions may be specified in terms of the rates of transitions in the derived transition system of the fluid components.

For this reason, in the context of an *immediate* GPEPA model G, we redefine the function $\mathcal{B}(G,H)$ to be the set of all *non-vanishing* derivative states of fluid components in component group H. Similarly, we redefine $\mathcal{B}(G)$ in terms of non-vanishing fluid component derivative states. An aggregated CTMC state of G can thus be represented, as in the case of a GPEPA model by a function $\mathsf{N} \in \mathcal{B}(G) \to \mathbb{Z}_+$. This means in turn that the (joint) component and apparent rate functions given for GPEPA models in Definitions 3.2.4, 3.4.1 and 3.2.6; and the (joint) derivative weighting function given in Definitions 3.2.5 and 3.4.4 need no modification

to be valid also for *i*GPEPA models as long as references to timed transitions between fluid component derivative states are interpreted in terms of their derived transition system. Furthermore, Theorems 3.2.7 and 3.4.3 which relate the (joint) component rate function to the aggregated rates of a GPEPA model remain valid for *i*GPEPA models when interpreted in terms of derived transition systems. First-moment and higher-moment component-counting ODEs for *i*GPEPA models can thus be constructed as per the definitions for GPEPA models (Definitions 3.3.1 and 3.4.5).

5.4.1 Action-counting ODEs from *i*GPEPA models

In order to implement the analysis of global passage times (from Section 4.3) using differential equations, we require an additional system of ODEs which approximate the number of transitions of a given action type which have occurred by some time $t \in \mathbb{R}_+$. We call these *action-counting ODEs*.

Action-counting ODEs are needed to count the number of signals emitted from a set of local probes by a given time. For some action type (timed or immediate) $a \in \mathcal{A}$, let $A_a(t)$ count the number of a-transitions which have occurred up to time t. We note that together with the component-counting processes, these processes determine an enlarged underlying state space and CTMC for an *i*GPEPA model.

We wish to define systems of ODEs which approximate moments of these processes in a similar manner to those defined in Sections 3.3 and 3.4 which approximate component-count moments. In order to approximate, say the variance of a particular action count, we require differential equations which depend upon joint moments of both component counts and action counts, such as $\mathbb{E}[N_{\boldsymbol{Client}}(t)A_{request}(t)]$ in the context of the client–server example model from Section 3.2.3. For this reason, a general moment $\mathbb{E}[M(t)]$ we are interested in is specified using a function $\mathsf{M} \in (\mathcal{B}(G) \cup \mathcal{A}) \to \mathbb{Z}_+$ by $M(t) := \prod_{B \in \mathcal{B}(G)} N_B(t)^{\mathsf{M}(B)} \prod_{a \in \mathcal{A}} A_a(t)^{\mathsf{M}(a)}$. We will write its ODE approximation as $v_{\mathsf{M}}(t)$ as in Section 3.4.

The aim of this section is to construct the differential equations for $v_{\mathsf{M}}(t)$, which encompass both component- and action-counting ODEs for *i*GPEPA. Readers who wish to pass over the derivation can find the resulting equation in Definition 5.4.4.

In order to construct these differential equations, we require a modified version of the joint evolution set (Definition 3.4.2), the derivative weighting function (Definition 3.2.5) and the joint derivative weighting function (Definition 3.4.4), given below. These alternate versions allow us to consider timed transitions between fluid component derivative states separately, conditioned on the sequence of immediate actions which may happen immediately after such a timed transition fires. Recall that this is recorded in the derived transition system by the sequence parameter $\mathcal{I}$ in $P \xrightarrow{(\alpha,r),\mathcal{I}} Q$.

This new version of the joint evolution set $\mathcal{J}(G)$ for an *i*GPEPA model G still consists of 3-tuples (J_-, J_+, α) but now the J_- set consists itself of tuples $(B, \mathcal{I})$ where $B \in \mathcal{B}(G)$ and $\mathcal{I}$ is a sequence of elements of $\mathcal{A}_i$. This allows the enumeration of the possible ways in which fluid component derivative states can evolve together with a particular action type, but treating each evolution separately depending on the immediate actions which may also be transmitted immediately after such an evolution. The formal definitions follow.

Definition 5.4.1 (*i*GPEPA:: Joint evolution set)**.** *Let G be an* i*GPEPA model. Then the joint evolution set is $\mathcal{J}(G)$, defined as follows.*

$$\begin{aligned}\mathcal{J}(M_1 \underset{L}{\bowtie} M_2) := &\{(J_-^1 \cup J_-^2, J_+^1 \cup J_+^2, \alpha) \,:\, (J_-^1, J_+^1, \alpha) \in \mathcal{J}(M_1),\, (J_-^2, J_+^2, \alpha) \in \mathcal{J}(M_2),\, \alpha \in L\}\\ &\cup \{(J_-, J_+, \alpha) \,:\, (J_-, J_+, \alpha) \in \mathcal{J}(M_1)\ ,\ \alpha \notin L\}\\ &\cup \{(J_-, J_+, \alpha) \,:\, (J_-, J_+, \alpha) \in \mathcal{J}(M_2)\ ,\ \alpha \notin L\}\\ \mathcal{J}(H\{D\}) := &\bigcup_{\substack{P \xrightarrow{(\alpha,\cdot),\mathcal{I}} Q\\ P,Q \in \mathcal{B}(G,H)}} \{\{((H,P),\mathcal{I})\},\{(H,Q)\},\alpha\}\end{aligned}$$

Definition 5.4.2 (*i*GPEPA:: Derivative weighting function)**.** *Let G be an* i*GPEPA model and $H \in \mathcal{G}(G)$ a component group label. Let $P, Q \in \mathcal{B}(G,H)$ be* i*PEPA fluid component derivative states and let $\alpha \in \mathcal{A}_t$ be a timed action type. Let also $\mathcal{I}$ be a sequence of immediate action types, that is, of elements of $\mathcal{A}_i$. Then:*

$$p_\alpha^{\mathcal{I}}(P,Q) := \frac{1}{r_\alpha(P)} \sum_{P \xrightarrow{(\alpha,\lambda),\mathcal{I}} Q} \lambda$$

This is defined to be zero when $r_\alpha(P) = 0$.

Definition 5.4.3 (*i*GPEPA:: Joint derivative weighting function)**.** *Let G be an* i*GPEPA model and let $(J_-, J_+, \alpha) \in \mathcal{J}(G)$. Then $\rho_\alpha(J_-, J_+)$ is the joint derivative weighting function for* i*GPEPA models and is defined as follows.*

$$\rho_\alpha(J_-, J_+) := \prod_{((H,P),\mathcal{I}) \in J_-} p_\alpha^{\mathcal{I}}(P, Q_{H,J_+})$$

where, as before, Q_{H,J_+} is defined as the unique fluid component derivative state Q, such that $(H,Q) \in J_+$.

Building directly on the approach of Section 3.4.2, we see that the action of the infinitesimal generator $\mathbf{A}_G$ of the underlying CTMC of G on $M(t)$ is given by:

$$\mathbf{A}_G\, M(t) = \sum_{(J_-,J_+,\alpha) \in \mathcal{J}(G)} \rho_\alpha(J_-,J_+)(F_+(t)F_-(t)F(t)H_+(t)H(t) - M(t))\mathcal{R}_\alpha(G, \mathsf{N}_t, J_-) \quad (5.9)$$

where the $F_\pm(t)$ and $F(t)$ are defined as in Section 3.4.2:[4]

$$F_\pm(t) := \prod_{B \in J_\pm \setminus J_\mp} (N_B(t) \pm 1)^{\mathsf{M}(B)} \qquad \text{and} \qquad F(t) := \prod_{B \in \mathcal{B}(G) \setminus (J_- \ominus J_+)} N_B(t)^{\mathsf{M}(B)}$$

and for $t \in \mathbb{R}_+$, $\mathsf{N}_t \in \mathcal{B}(G) \to \mathbb{Z}_+$ is given by $\mathsf{N}_t(B) := N_B(t)$ for all $B \in \mathcal{B}(G)$. The new quantities $H_+(t)$ and $H(t)$ are defined similarly but to capture changes in the *action counts*:

$$H_+(t) := \prod_{a \in \mathcal{I}(J_-) \cup \{\alpha\}} (A_a(t) + \mathcal{C}_a(J_-))^{\mathsf{M}(a)} \qquad \text{and} \qquad H(t) := \prod_{a \notin \mathcal{I}(J_-) \cup \{\alpha\}} A_a(t)^{\mathsf{M}(a)}$$

[4] When considering elements $(J_-, J_+, \alpha) \in \mathcal{J}(G)$ in terms of the new definition of the joint evolution set for *i*GPEPA models (Definition 5.4.1), for $B \in \mathcal{B}(G)$, we will write $B \in J_-$ to mean $(B, \cdot) \in J_-$ for brevity and to maintain consistency with Chapter 3.

where the set $\mathcal{I}(J_-) := \cup_{(B,\mathcal{I})\in J_-}\mathcal{I}$ and, for immediate action types $\mathsf{a} \in \mathcal{A}_i$, $\mathcal{C}_{\mathsf{a}}(J_-)$ is the total number of times a occurs in sequences in $\{\mathcal{I} : (B,\mathcal{I}) \in J_-\}$. For timed action types $\alpha \in \mathcal{A}_t$, $\mathcal{C}_\alpha(J_-) := 1$. Continuing to follow Section 3.4.3, we may apply the binomial theorem to expand the quantity $F_+(t)F_-(t)F(t)H_+(t)H(t)$ in Equation 5.9 into the sum of all monomials:

$$\prod_{B\in J_-\setminus J_+}(-1)^{\mathsf{M}(B)-\mathsf{K}(B)}\prod_{B\in\mathcal{B}(G)}\binom{\mathsf{M}(B)}{\mathsf{K}(B)}N_B(t)^{\mathsf{K}(B)}\prod_{a\in\mathcal{A}}\binom{\mathsf{M}(a)}{\mathsf{K}(a)}\mathcal{C}_a(J_-)^{\mathsf{M}(a)-\mathsf{K}(a)}A_a(t)^{\mathsf{K}(a)} \quad (5.10)$$

taken over all elements $\mathsf{K} \in (\mathcal{B}(G)\cup\mathcal{A}) \to \mathbb{Z}_+$, which satisfy $0 \le \mathsf{K}(B) \le \mathsf{M}(B)$ for all $B \in J_-\ominus J_+$, $\mathsf{K}(B) = \mathsf{M}(B)$ for all $B \in \mathcal{B}(G)\setminus(J_-\ominus J_+)$; $0 \le \mathsf{K}(a) \le \mathsf{M}(a)$ for all $a \in \mathcal{I}(J_-)\cup\{\alpha\}$ and $\mathsf{K}(a) = \mathsf{M}(a)$ for all $a \notin \mathcal{I}(J_-)\cup\{\alpha\}$.

Now consider the term $F_+(t)F_-(t)F(t)H_+(t)H(t) - M(t)$ and note that $M(t)$ simply serves to cancel the highest-order such monomial, i.e. the case of $\mathsf{K}(B) = \mathsf{M}(B)$ for all $B \in \mathcal{B}(G)$ and $\mathsf{K}(a) = \mathsf{M}(a)$ for all $a \in \mathcal{A}$. Therefore, after this cancellation, all monomials in the sum will have order at most $M-1$, where $M := \sum_{B\in\mathcal{B}(G)}\mathsf{M}(B) + \sum_{a\in\mathcal{A}}\mathsf{M}(a)$ is the order of the moment of interest. That is, $F_+(t)F_-(t)F(t)H_+(t)H(t) - M(t)$ is equal to the sum of all monomials of the form of Equation 5.10 taken over all elements $\mathsf{K} \in (\mathcal{B}(G)\cup\mathcal{A}) \to \mathbb{Z}_+$, which satisfy $0 \le \mathsf{K}(B) \le \mathsf{M}(B)$ for all $B \in J_-\ominus J_+$, $\mathsf{K}(B) = \mathsf{M}(B)$ for all $B \in \mathcal{B}(G)\setminus(J_-\ominus J_+)$; $0 \le \mathsf{K}(a) \le \mathsf{M}(a)$ for all $a \in \mathcal{I}(J_-)\cup\{\alpha\}$ and $\mathsf{K}(a) = \mathsf{M}(a)$ for all $a \notin \mathcal{I}(J_-)\cup\{\alpha\}$; where $\sum_{B\in\mathcal{B}(G)}\mathsf{K}(B)+\sum_{a\in\mathcal{A}}\mathsf{K}(a) \neq M$. We denote this subset of $(\mathcal{B}(G)\cup\mathcal{A}) \to \mathbb{Z}_+$ by $\mathcal{K}_\alpha(J_-, J_+, \mathsf{M})$.

Then adapting the reasoning from Section 3.4.3 in a straightforward manner, we obtain the following ODE approximating the joint component-count and action-count moment specified by M.

Definition 5.4.4 (*i*GPEPA:: Higher-order moment ODEs)**.** *Let G be an* i*GPEPA model. We define the evolution of the $v_{\mathsf{M}}(t)$ over time for $\mathsf{M} \in (\mathcal{B}(G)\cup\mathcal{A}) \to \mathbb{Z}_+$ by the system of first-order coupled ODEs:*

$$\dot{v}_{\mathsf{M}}(t) = \sum_{(J_-,J_+,\alpha)\in\mathcal{J}(G)}\rho_\alpha(J_-,J_+)\sum_{\mathsf{K}\in\mathcal{K}_\alpha(J_-,J_+,\mathsf{M})}\left(\left(\prod_{B\in J_-\setminus J_+}(-1)^{\mathsf{M}(B)-\mathsf{K}(B)}\prod_{B\in\mathcal{B}(G)}\binom{\mathsf{M}(B)}{\mathsf{K}(B)}\prod_{a\in\mathcal{A}}\binom{\mathsf{M}(a)}{\mathsf{K}(a)}\mathcal{C}_a(J_-)^{\mathsf{M}(a)-\mathsf{K}(a)}\right)\mathcal{R}_\alpha(G, V_t^*[\mathsf{K}], J_-)\right) \quad (5.11)$$

where, like in Section 3.4.3, for $t \in \mathbb{R}_+$, $V_t^*[\mathsf{K}] \in \mathcal{B}(G) \to \mathbb{R}_+$ is given by, for all $B \in \mathcal{B}(G)$:

$$V_t^*[\mathsf{K}](B) := v_{\mathsf{J}}(t)$$

where $\mathsf{J}(a) = \mathsf{K}(a)$ for all $a \in \mathcal{A}$, $\mathsf{J}(B') = \mathsf{K}(B')$ for all $B' \in \mathcal{B}(G)$ with $B' \neq B$ and $\mathsf{J}(B) = \mathsf{K}(B) + 1$. Note that since $\sum_{B'\in\mathcal{B}(G)}\mathsf{K}(B') + \sum_{a\in\mathcal{A}}\mathsf{K}(a) \le M-1$ for all $\mathsf{K} \in \mathcal{K}_\alpha(J_-, J_+, \mathsf{M})$, ODE approximations of joint moments of order at most M occur in such terms.

The initial condition for each moment, $\mathsf{M} \in (\mathcal{B}(G) \cup \mathcal{A}) \to \mathbb{Z}_+$*, is that naturally implied by the initial state of* G*. Specifically, we define:*

$$v_{\mathsf{M}}(0) := \begin{cases} \prod_{B \in \mathcal{B}(G)} \mathcal{C}(G, B)^{\mathsf{M}(B)} & : \mathsf{M}(a) = 0 \text{ for all } a \in \mathcal{A} \\ 0 & : \text{otherwise} \end{cases}$$

We have assumed here that there are no fluid components in the model with vanishing initial states, so all action counts are initially zero. The initial conditions for moments involving immediate action counts would need to be modified to account for immediate transitions which happen immediately at time $t = 0$ *if there were fluid components in* G *with vanishing initial states.*

When we are only interested in the ODEs approximating *first moments* of component counts and of action counts, we will write the former as $v_{H,P}(t)$ for $(H, P) \in \mathcal{B}$ and the latter as $w_a(t)$ for $a \in \mathcal{A}$. In particular, this is the notation we will adopt in Chapter 6 where we discuss the convergence of the first-moment approximations.

5.5 Differential-equation approximation of probe-specified individual passage times

In this section, we will consider *individual* passage-time measures specified using unified stochastic probes. We will see how the use of stochastic probes allows us to automate the local state-space modifications which were necessary in the previous chapter to specify passage-time measurements. Furthermore, we will show how such performance queries can be analysed efficiently using the kinds of ODE-based techniques introduced in Section 4.4.

A probe-specified individual passage-time measurement is defined by one or more nested local probes attached to one specific fluid component within an *i*GPEPA model. The associated global probe has two possible fixed forms which distinguish between a transient and a steady-state individual passage-time measurement, respectively:

$$\mathit{Probe}_g ::= \mathsf{begin} : \mathsf{start}, \mathsf{end} : \mathsf{stop} \tag{5.12}$$

$$\mid \mathsf{begin} : \mathsf{start}, \mathsf{end} : \mathsf{stop}^{\hookleftarrow} \tag{5.13}$$

The signals begin and end delegate control of the passage time to the local probes monitoring the individual component of interest. For ease of presentation we will assume that the individual being monitored is the first of a number of identical fluid components which together are the only members of a single component group. This is not a real restriction since additional component groups can be added if necessary. However, it also underscores the fact that the ODE approximation is more accurate when applied to individuals who are part of a large population of components.

In the sections which follow, we show how ODE-based approaches can be applied to approximate the distribution of both types of individual passage-time measures. Note that it is not necessary to consider the case of nested local probes explicitly in the following sections because the

composition of a local probe with a fluid component results in another fluid component. Thus, from the point of view of the analysis, it does not matter how many local probes were used originally to form the resulting probed fluid component.

5.5.1 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:

$$\begin{aligned} &\mathsf{begin} : \mathsf{start}, \mathsf{end} : \mathsf{stop}^{\hookleftarrow} \\ &\mathsf{observes} \quad Pb \stackrel{def}{=} R_l : \mathsf{begin}, R_l : \mathsf{end}^{\hookleftarrow} \\ &\mathsf{where} \quad H\{P[?n]\} \Longrightarrow H\{(P \underset{*}{\bowtie} Pb) \wr\wr P[?n-1]\} \\ &\mathsf{in} \quad G \end{aligned} \tag{5.14}$$

The location specification applies the local probe Pb to the first fluid component in a group H of identical components since we are interested in monitoring a single individual. For example, the steady-state individual passage-time measurement constructed manually in Section 4.4.1, *the time taken for a client to complete one cycle*, could be expressed using the following probe specification:

$$\begin{aligned} &\mathsf{begin} : \mathsf{start}, \mathsf{end} : \mathsf{stop}^{\hookleftarrow} \\ &\mathsf{observes} \quad Pb \stackrel{def}{=} \epsilon : \mathsf{begin}, \mathit{think} : \mathsf{end}^{\hookleftarrow} \\ &\mathsf{where} \quad \mathbf{Clients}\{\boldsymbol{Client}[?n]\} \Longrightarrow \mathbf{Clients}\{(\boldsymbol{Client} \underset{*}{\bowtie} Pb) \wr\wr \boldsymbol{Client}[?n-1]\} \\ &\mathsf{in} \quad \boldsymbol{CS}(n, m) \end{aligned} \tag{5.15}$$

In order to compute the CDF of such a passage time, we will need to consider two probed versions of the model to be queried G. $\tilde{G}$ represents the model G in composition with Pb after application of the location specification in Equation 5.14. Analysis of $\tilde{G}$ will capture the steady-state distribution which will be used to initialise 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 $Pb' \stackrel{def}{=} R_l : \mathsf{begin}, R_l : \mathsf{end}$. The model $\tilde{G}'$ will be used to capture the evolution of the passage time itself. Recall that in order to perform the analysis of the passage time specified by Equation 5.15 in Section 4.4.1 of the previous chapter, we had to manually construct a modified version of the ***Client*** component, repeated in Figure 5.5 for emphasis. This component is now obtained automatically by the composition of the ***Client*** component with the absorbing local probe $\epsilon : \mathsf{begin}, \mathit{think} : \mathsf{end}$. Automating this otherwise complex model transformation has the dual benefit of removing a source of error and making the relationship to the behavioural passage specification explicit.

Our analysis approach is similar to that of Section 4.4.1. However we deviate from this slightly in order to handle correctly the case of multiple local passage-time source states in the observed

[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.

$$
\begin{aligned}
\boldsymbol{Client} &\stackrel{def}{=} (request, r_r).\boldsymbol{Client_{waiting}} \\
\boldsymbol{Client_{waiting}} &\stackrel{def}{=} (data, r_d).\boldsymbol{Client_{think}} + (timeout, r_{tmt}).\boldsymbol{Client} \\
\boldsymbol{Client_{think}} &\stackrel{def}{=} (think, r_t).\boldsymbol{Client'} \\
\boldsymbol{Client'} &\stackrel{def}{=} (request, r_r).\boldsymbol{Client_{waiting}}' \\
\boldsymbol{Client_{waiting}}' &\stackrel{def}{=} (data, r_d).\boldsymbol{Client_{think}}' + (timeout, r_{tmt}).\boldsymbol{Client'} \\
\boldsymbol{Client_{think}}' &\stackrel{def}{=} (think, r_t).\boldsymbol{Client'}
\end{aligned}
$$

Figure 5.5: Modified ***Client*** component for the computation of the steady-state individual passage-time query of Section 4.4.1.

fluid component. In order to achieve this, we consider explicitly ODEs which capture the probability distribution of the probed component, whereas, in Section 4.4.1, these quantities were obtained indirectly by suitable normalisation of the total component-count expectations.

As in the previous chapter, we consider the model $\tilde{G}$ evolving in the stationary regime and then at some fixed time its evolution is switched to that of $\tilde{G}'$, initialised by the current state of $\tilde{G}$. Then by timing how long it takes for the absorbing probe Pb' in $\tilde{G}'$ to enter its absorbed state, say, Pb_{Acc}, we can measure the duration of the passage time of interest. Define $C(t) \in ds^*(P \underset{*}{\bowtie} Pb')$ to be the stochastic process, which tracks the state of the probed fluid component in the group H as the model $\tilde{G}'$ evolves. Let time 0 be the arbitrary fixed time at which the evolution of the model switches to $\tilde{G}'$, initialised by the state of $\tilde{G}$. Then the quantity we wish to compute is:

$$\mathbb{P}\{C(t) \in — \underset{*}{\bowtie} Pb_{\text{Acc}} \mid C(0) \in \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb), C(0-) \notin \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb)\} \tag{5.16}$$

where $— \underset{*}{\bowtie} Pb_{\text{Acc}}$ is shorthand for the set of all $Q \underset{*}{\bowtie} Pb_{\text{Acc}} \in ds^*(P \underset{*}{\bowtie} Pb')$ and Pb_{Acc} is the absorbing accepting state of the local probe Pb'. Also, $\mathcal{S}_{\mathsf{a}}(P) := \{P' \in ds^*(P) : \exists P'', \text{ st. } P'' \xrightarrow{(\cdot,\cdot),(\ldots,\mathsf{a},\ldots)} P'\}$ is the *one-step successor function*, which returns all derivative states reachable from P by a transition which fires an immediate action of type $\mathsf{a} \in \mathcal{A}_i$.

As in Section 4.4.1, the first step in the procedure is to approximate, by the ODEs of Definition 3.3.1, the stationary expectations of the component counts of the model $\tilde{G}$. As before, we achieve this by taking the limit of the ODE solutions as $t \to \infty$, either by numerical integration or direct computation of a fixed point.[6] For any $(Y, Q) \in \mathcal{B}(\tilde{G})$, we will write $\tilde{v}_{Y,Q}$ for the approximation to $\tilde{N}_{Y,Q} := \lim_{t\to\infty} \mathbb{E}[\tilde{N}_{Y,Q}(t)]$ obtained in this manner. Since the ODE solutions approximate component-count expectations and there is a single probed fluid component in the model $\tilde{G}$, we see that the quantities corresponding to non-vanishing derivative states of the probed fluid component, $\tilde{v}_{H,R}$ for $R \in ds^*(P \underset{*}{\bowtie} Pb)$, approximate the steady-state probability that the probed component is in the given state.[7]

[6] Recall from the last chapter that the system of ODEs usually has only one *meaningful* fixed point. As mentioned earlier, this is discussed in more detail in Chapter 6 where conditions guaranteeing convergence of the stationary distribution to the ODE's fixed point for increasing component population size are given.

[7] It may initially seem surprising that this will provide a good approximation since there is only one copy of the fluid component $P \underset{*}{\bowtie} Pb$ in the model. However, the key point which ensures that the approximation is good is that this component exists within a large population of identically-distributed P-components (the

However, in order to capture the passage-time quantity of Equation 5.16 correctly, we need to compute the steady-state distribution of the probed fluid component *given that the* begin *signal has just fired*. Computing probabilities conditioned on transition instants for a subset of *selected transitions* in this manner is an application of *Palm calculus* applied to stationary continuous-time Markov chains (see, for example, the book by Le Boudec [178, Section 7.5] or Serfozo [218, Section 4.16]). In particular, the probability that the probed component is in state $R \in ds^*(P \underset{*}{\bowtie} Pb)$ 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 R, divided by the total expected rate of all transitions firing begin signals [e.g. 218, Definition 96 and Proposition 97], that is:

$$\mathbb{P}\{C(0) = R \mid C(0) \in \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb), C(0-) \notin \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb)\} = \frac{\sum_{\alpha \in \mathcal{A}_t} \sum_{C \in ds^*(P \underset{*}{\bowtie} Pb)} \left(\frac{1}{r_\alpha(C)} \sum_{C \xrightarrow{(\alpha,\lambda),(\ldots,\mathsf{begin},\ldots)} R} \lambda \times \mathbb{E}\left[\mathcal{R}_\alpha(\tilde{G}, \tilde{\mathsf{N}}_0, H, C)\right]\right)}{\sum_{\alpha \in \mathcal{A}_t} \sum_{C \in ds^*(P \underset{*}{\bowtie} Pb)} \left(\frac{1}{r_\alpha(C)} \sum_{C \xrightarrow{(\alpha,\lambda),(\ldots,\mathsf{begin},\ldots)}} \lambda \times \mathbb{E}\left[\mathcal{R}_\alpha(\tilde{G}, \tilde{\mathsf{N}}_0, H, C)\right]\right)}$$

where, similarly to before, $\tilde{\mathsf{N}}_0 \in \mathcal{B}(\tilde{G}) \to \mathbb{Z}_+$ represents the state of the underlying CTMC of the model $\tilde{G}$ in the stationary regime at time 0. The ODE approximation to this expression is obtained by substituting the ODE approximations to the stationary component counts obtained above in place of $\tilde{\mathsf{N}}_0$. Specifically, defining $\tilde{\mathsf{V}} \in \mathcal{B}(\tilde{G}) \to \mathbb{R}_+$ by $\tilde{\mathsf{V}}(Y, Q) := \tilde{v}_{Y,Q}$ for all $(Y, Q) \in \mathcal{B}(\tilde{G})$, we obtain the ODE approximation:

$$\mathbb{P}\{C(0) = R \mid C(0) \in \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb), C(0-) \notin \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb)\} \approx \frac{\sum_{\alpha \in \mathcal{A}_t} \sum_{C \in ds^*(P \underset{*}{\bowtie} Pb)} \left(\frac{1}{r_\alpha(C)} \sum_{C \xrightarrow{(\alpha,\lambda),(\ldots,\mathsf{begin},\ldots)} R} \lambda \times \mathcal{R}_\alpha(\tilde{G}, \tilde{\mathsf{V}}, H, C)\right)}{\sum_{\alpha \in \mathcal{A}_t} \sum_{C \in ds^*(P \underset{*}{\bowtie} Pb)} \left(\frac{1}{r_\alpha(C)} \sum_{C \xrightarrow{(\alpha,\lambda),(\ldots,\mathsf{begin},\ldots)}} \lambda \times \mathcal{R}_\alpha(\tilde{G}, \tilde{\mathsf{V}}, H, C)\right)} \tag{5.17}$$

Finally, in order to compute the passage time of interest, the ODEs are constructed according to Definition 3.3.1 for the absorbing model $\tilde{G}'$. For each $(Y, Q) \in \mathcal{B}(\tilde{G}')$, we will write $\tilde{v}'_{Y,Q}(t)$ for their solutions. In order to capture correctly the passage-time quantity starting from a transition instant, the initial condition $\tilde{v}'_{H,R}(0)$ for a probed component derivative state $R \in ds^*(P \underset{*}{\bowtie} Pb')$ must be the conditioned quantity computed above in Equation 5.17 (and is zero if $R \notin ds^*(P \underset{*}{\bowtie} Pb)$). For all other $(Y, Q) \in \mathcal{B}(\tilde{G}')$, the initial value $\tilde{v}'_{Y,Q}(0)$ is set to $\tilde{v}_{Y,Q}$. The ODEs for $\tilde{G}'$ can then be integrated numerically and the passage-time CDF of Equation 5.16 is approximated by $\sum_{C \in — \underset{*}{\bowtie} Pb_{\mathrm{Acc}}} \tilde{v}'_{H,C}(t)$.

We note finally that, given a probe-specified steady-state individual passage-time measurement of the form of Equation 5.14, the entire process described above culminating in the approximation of the corresponding CDF could be automated. This includes the translation of the local probe to a DFA and subsequently to an *i*PEPA component (Section 5.3.2), the composition of the component with the model to be observed and the construction of its non-vanishing state space (Section 5.2.1). Furthermore, the derivation of the system of ODEs corresponding to the probed *i*GPEPA model is a mechanical task and simply consists of applying Definition 3.3.1. The CDF is then approximated simply by evaluating the appropriate functional expressions of the ODE solutions as documented above.

only difference being that this one component is probed). This argument will be given formally in the proof of Theorem 6.4.2 in Chapter 6.

At present, the GPA tool [229, 235] mentioned earlier in this thesis does not support all of the above steps. Specifically, the translation to DFA and composition of the local probe with the model to be observed must currently be achieved manually. However, it is planned that GPA will be able to take measure specifications such as Equation 5.14 as input and return a CDF computed both using traditional approaches, where feasible, and the ODE-based approach given above.

5.5.2 Transient individual passage times

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

$$\begin{aligned} &\mathsf{begin} : \mathsf{start}, \mathsf{end} : \mathsf{stop} \\ &\mathsf{observes} \quad Pb \stackrel{def}{=} R_l : \mathsf{begin}, R_l : \mathsf{end} \\ &\mathsf{where} \quad H\{P[?n]\} \Longrightarrow H\{(P \underset{*}{\bowtie} Pb) \wr\wr P[?n-1]\} \\ &\mathsf{in} \quad G \end{aligned} \tag{5.18}$$

As in the previous section, the location specification applies the local probe Pb to the first fluid component in a group H of identical components since we are interested in monitoring a single individual. For example, the transient individual passage-time measurement constructed manually in Section 4.4.2, *the time taken for a client to complete its second cycle*, could be expressed using the following probe specification:

$$\begin{aligned} &\mathsf{begin} : \mathsf{start}, \mathsf{end} : \mathsf{stop} \\ &\mathsf{observes} \quad Pb \stackrel{def}{=} \mathit{think} : \mathsf{begin}, \mathit{think} : \mathsf{end} \\ &\mathsf{where} \quad \mathbf{Clients}\{\boldsymbol{Client}[?n]\} \Longrightarrow \mathbf{Clients}\{(\boldsymbol{Client} \underset{*}{\bowtie} Pb) \wr\wr \boldsymbol{Client}[?n-1]\} \\ &\mathsf{in} \quad \boldsymbol{CS}(n, m) \end{aligned} \tag{5.19}$$

Since we do not need to compute a stationary distribution in this section, we need only consider one probed version of the model, obtained by composition with the absorbing local probe Pb as specified by Equation 5.18. We call the composed model $\tilde{G}$. In the case of the passage time specified by Equation 5.19, the composed fluid component automates the construction of the modified ***Client*** component (repeated in Figure 5.6) which had to be constructed manually in Section 4.4.2.

In order to approximate the passage time, we will mostly follow the approach given in Section 4.4.2 of the previous chapter. As in the last section, we will however deviate from that approach in order to handle correctly the case of multiple local start states in the fluid component under observation. Let T_{begin} be the random time at which the begin signal is transmitted by the fluid component under observation and let T_{end} be the random time at which the end signal is transmitted. Let $F(t) := \mathbb{P}\{T_{\mathsf{end}} - T_{\mathsf{begin}} \leq t\}$ be the cumulative distribution function (CDF) of interest. Then let $F_s(t) := \mathbb{P}\{T_{\mathsf{end}} - T_{\mathsf{begin}} \leq t \mid T_{\mathsf{begin}} = s\}$ and $g(s) := \frac{\mathrm{d}}{\mathrm{d}s}G(s)$ where $G(s) := \mathbb{P}\{T_{\mathsf{begin}} \leq s\}$. Then we have:

$$F(t) = \int_0^\infty F_s(t) g(s)\, \mathrm{d}s \tag{5.20}$$

$$
\begin{aligned}
\boldsymbol{Client} &\stackrel{def}{=} (request, r_r).\boldsymbol{Client_{waiting}} \\
\boldsymbol{Client_{waiting}} &\stackrel{def}{=} (data, r_d).\boldsymbol{Client_{think}} + (timeout, r_{tmt}).\boldsymbol{Client} \\
\boldsymbol{Client_{think}} &\stackrel{def}{=} (think, r_t).\widehat{\boldsymbol{Client}} \\
\widehat{\boldsymbol{Client}} &\stackrel{def}{=} (request, r_r).\widehat{\boldsymbol{Client_{waiting}}} \\
\widehat{\boldsymbol{Client_{waiting}}} &\stackrel{def}{=} (data, r_d).\widehat{\boldsymbol{Client_{think}}} + (timeout, r_{tmt}).\widehat{\boldsymbol{Client}} \\
\widehat{\boldsymbol{Client_{think}}} &\stackrel{def}{=} (think, r_t).\boldsymbol{Client}' \\
\boldsymbol{Client}' &\stackrel{def}{=} (request, r_r).\boldsymbol{Client_{waiting}}' \\
\boldsymbol{Client_{waiting}}' &\stackrel{def}{=} (data, r_d).\boldsymbol{Client_{think}}' + (timeout, r_{tmt}).\boldsymbol{Client}' \\
\boldsymbol{Client_{think}}' &\stackrel{def}{=} (think, r_t).\boldsymbol{Client}'
\end{aligned}
$$

Figure 5.6: Modified ***Client*** component for the computation of the transient individual passage-time query of Section 4.4.2.

The cumulative distribution function for the transient individual passage time can be computed directly from Equation 5.20 once we have access to the quantities $F_s(t)$ and $g(s)$.

Approximation of $F_s(t)$

As before, define $C(t) \in ds^*(P \underset{*}{\bowtie} Pb)$ to be the stochastic process, which tracks the state of the probed fluid component in the group H as the model $\tilde{G}$ evolves. For $(Y, Q) \in \mathcal{B}(\tilde{G})$, write $\tilde{v}_{Y,Q}(t)$ for the solutions of the ODEs constructed according to Definition 3.3.1 with the initial conditions given by the initial state of $\tilde{G}$. We begin by noting that:

$$F_s(t) = \mathbb{P}\{C(s+t) \in -- \underset{*}{\bowtie} Pb_{\text{Acc}} \mid C(s) \in \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb), C(s-) \notin \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb)\}$$

Thus in order to capture this quantity correctly, we must compute the distribution of the probed fluid component *given that the* begin *signal has just fired at time* s. The appropriate conditional distribution on states $R \in ds^*(P \underset{*}{\bowtie} Pb)$ can, similarly to the last section, be computed by employing results from the theory of time-dependent Palm calculus [e.g. 218, Definition 98 and Proposition 97]:

$$
\begin{aligned}
&\mathbb{P}\{C(s) = R \mid C(s) \in \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb), C(s-) \notin \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb)\} = \\
&\qquad \frac{\sum_{\alpha \in \mathcal{A}_t} \sum_{C \in ds^*(P \underset{*}{\bowtie} Pb)} \left(\frac{1}{r_\alpha(C)} \sum_{C \xrightarrow{(\alpha,\lambda),(\ldots,\mathsf{begin},\ldots)} R} \lambda \times \mathbb{E}\left[\mathcal{R}_\alpha(\tilde{G}, \tilde{\mathsf{N}}_s, H, C)\right] \right)}{\sum_{\alpha \in \mathcal{A}_t} \sum_{C \in ds^*(P \underset{*}{\bowtie} Pb)} \left(\frac{1}{r_\alpha(C)} \sum_{C \xrightarrow{(\alpha,\lambda),(\ldots,\mathsf{begin},\ldots)}} \lambda \times \mathbb{E}\left[\mathcal{R}_\alpha(\tilde{G}, \tilde{\mathsf{N}}_s, H, C)\right] \right)}
\end{aligned}
$$

where $\tilde{\mathsf{N}}_s \in \mathcal{B}(\tilde{G}) \to \mathbb{Z}_+$ represents the state of the underlying CTMC of the model $\tilde{G}$ at time s. As in the previous section, the ODE approximation to this expression is obtained by substituting the ODE approximations to the component counts in place of $\tilde{\mathsf{N}}_s$. Specifically, defining $\tilde{\mathsf{V}}_s \in \mathcal{B}(\tilde{G}) \to \mathbb{R}_+$ by $\tilde{\mathsf{V}}_s(H, P) := \tilde{v}_{H,P}(s)$ for all $(Y, Q) \in \mathcal{B}(\tilde{G})$, we obtain the ODE

approximation:

$$\mathbb{P}\{C(s) = R \mid C(s) \in \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb), C(s-) \notin \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb)\} \approx \frac{\sum_{\alpha \in \mathcal{A}_t} \sum_{C \in ds^*(P \underset{*}{\bowtie} Pb)} \left(\frac{1}{r_\alpha(C)} \sum_{C \xrightarrow{(\alpha,\lambda),(\ldots,\mathsf{begin},\ldots)} R} \lambda \times \mathcal{R}_\alpha(\tilde{G}, \tilde{\mathsf{V}}_s, H, C) \right)}{\sum_{\alpha \in \mathcal{A}_t} \sum_{C \in ds^*(P \underset{*}{\bowtie} Pb)} \left(\frac{1}{r_\alpha(C)} \sum_{C \xrightarrow{(\alpha,\lambda),(\ldots,\mathsf{begin},\ldots)}} \lambda \times \mathcal{R}_\alpha(\tilde{G}, \tilde{\mathsf{V}}_s, H, C) \right)} \tag{5.21}$$

Then, in order to compute the quantity $F_s(t)$, we construct a different solution to the system of ODEs derived from $\tilde{G}$. We will write this solution as $v^s_{Y,Q}(t)$ for $(Y, Q) \in \mathcal{B}(\tilde{G})$ since its initial conditions will be defined in terms of the quantities $v_{Y,Q}(s)$. Specifically, similarly to the previous section, the initial condition $\tilde{v}^s_{H,R}(0)$ for a probed component derivative state $R \in ds^*(P \underset{*}{\bowtie} Pb)$ is defined as the conditioned quantity computed above in Equation 5.21. For all other $(Y, Q) \in \mathcal{B}(\tilde{G})$, the initial value $\tilde{v}^s_{Y,Q}(0)$ is set to $\tilde{v}_{Y,Q}(s)$. This allows us to approximate the conditioned CDF by $F_s(t) \approx \sum_{C \in -\underset{*}{\bowtie} Pb_{\mathrm{Acc}}} \tilde{v}^s_{H,C}(t)$.

Approximation of $g(s)$

In order to approximate $g(s)$, we first obtain the approximation $G(s) \approx \sum_{Q \in \mathcal{Q}} \tilde{v}_{H,Q}(s)$ where $\mathcal{Q} := \bigcup_{Q \in \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb)} ds^*(Q)$ is the set of all states that $P \underset{*}{\bowtie} Pb$ can be in *after the* begin signal has been transmitted. Differentiating then provides the approximation of $g(s)$.

Approximation of the CDF $F(t)$

The cumulative distribution function of interest can then be computed directly from Equation 5.20 by substituting in the appropriate ODE approximations.

Like in the previous section, given a probe-specified transient individual passage-time measurement of the form of Equation 5.18, the entire approximation process described above could be automated and support for direct evaluation of such measures is planned for the GPA tool [229, 235].

5.6 Differential-equation approximation of probe-specified global passage times

Global passage-time measurements consider the joint behaviour of an entire population of fluid components. Local behaviour is picked out by first attaching local probes to fluid components and the simultaneous evolution of many fluid components in the model is then considered by specifying a suitable global probe. In this section, we show how such passage-time measures can be approximated using the ODE-based point-mass approximation technique of Section 4.3.

A transient global passage-time measurement on the model G is encoded by a probe specification

of the form:

$$\begin{array}{l} \epsilon : \mathsf{start}, R_g : \mathsf{stop} \\ \mathsf{observes} \quad \{Probe_l\} \\ \mathsf{where} \quad \{Location\} \\ \mathsf{in} \quad G \end{array} \tag{5.22}$$

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 — they may use the entire grammar of Equation 5.4. However, for the point-mass approximation convergence result of Section 6.4.1 to be stated, a local probe must be applied to each component in a component group (or not at all). Therefore, the location specifications in $\{Location\}$ must each have the form:

$$H\{P[?n]\} \Longrightarrow H\{(P \underset{*}{\bowtie} Probe_l)[?n]\}$$

for some $(H, P) \in \mathcal{B}(G)$ and local probe $Probe_l$.

Also in order for the point-mass approximation to be applicable, it is necessary to consider a restricted form of the grammar of Equation 5.6 which specifies the form of the regular expression R_g in the global probe. Specifically, when ODE-analysis techniques are to be applied, we consider the restricted grammar:

$$R_g ::= \{pred\}R_g \;\mid\; R_g, R_g \;\mid\; R_g \,|\, R_g \;\mid\; R_g; R_g \;\mid\; R_g^a[n] \tag{5.23}$$

where:

$$R_g^a ::= R_g^a \,|\, R_g^a \;\mid\; action$$

and $action \in \mathcal{A} \setminus \{\tau\}$ is a non-hidden action (or signal) type.

It should be noted that global passage times where R_g falls outside of this pattern can still be analysed as long as the state space is small enough. In such cases, a general global probe (as defined by the complete grammar of Equation 5.6) can be translated to an iGPEPA component following Section 5.3.2 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.

In our restricted grammar, we disallow the observation of a single action or signal, instead allowing fundamental observations of the form $(action_1 \mid \ldots \mid action_k)[n]$. The reason for requiring blocks of n observations of single actions or signals is that in order to prove convergence to point mass results, we will require that the number of such observed actions increases proportional to the population size. We do not allow the *sequence*, *both* or other constructions at this level of the grammar. To see why, consider for example, the regular expression $(a, b)[n]$. This matches when it observes n pairs of a and b actions in that order. Such a measurement taken at the global level is not amenable to ODE-based analysis since the scaling regime considered in this thesis does not in general allow for the resolution of questions regarding the ordering of individual, globally-observed actions.

We also do not allow observations of the form $\overline{action}$ since passage-time random variables specified in terms of such observations made globally are pathological in the large population

limit. This is because specific uninterrupted sequences of actions observed globally become increasingly unlikely as the component populations are scaled up.

The first step in the ODE analysis of a probe-specified global passage time is to apply the local probes in accordance with the location specifications for the measurement (as given in Equation 5.22). We assume that this has been done and that we are left with a model that contains the integrated local probes, say $\tilde{G}$. In order to apply the ODE point-mass approximation, we proceed by translating the global probe to an equivalent symbolic mathematical expression in terms of the action and component counts. Specifically, the function $\mathcal{U}(\tilde{G}, \mathsf{N}_t, \mathsf{A}_t, R_g, 0)$ provides an automatic translation of the global probe to an equivalent algebraic expression in terms of the component-counting and action-counting stochastic processes encoded by $\mathsf{N}_t \in \mathcal{B}(\tilde{G}) \to \mathbb{Z}_+$ and $\mathsf{A}_t \in \mathcal{A} \to \mathbb{Z}_+$, respectively:

$$\begin{aligned}
\mathcal{U}(\tilde{G}, \mathsf{N}_t, \mathsf{A}_t, (R_g^1, R_g^2), e) &:= \mathcal{U}(\tilde{G}, \mathsf{N}_t, \mathsf{A}_t, R_g^2, \mathcal{U}(\tilde{G}, \mathsf{N}_t, \mathsf{A}_t, R_g^1, e)) \\
\mathcal{U}(\tilde{G}, \mathsf{N}_t, \mathsf{A}_t, (R_g^1 \mid R_g^2), e) &:= \min(\mathcal{U}(\tilde{G}, \mathsf{N}_t, \mathsf{A}_t, R_g^1, e), \mathcal{U}(\tilde{G}, \mathsf{N}_t, \mathsf{A}_t, R_g^2, e)) \\
\mathcal{U}(\tilde{G}, \mathsf{N}_t, \mathsf{A}_t, (R_g^1; R_g^2), e) &:= \max(\mathcal{U}(\tilde{G}, \mathsf{N}_t, \mathsf{A}_t, R_g^1, e), \mathcal{U}(\tilde{G}, \mathsf{N}_t, \mathsf{A}_t, R_g^2, e)) \\
\mathcal{U}(\tilde{G}, \mathsf{N}_t, \mathsf{A}_t, \{pred\}R_g, e) &:= \mathcal{U}(\tilde{G}, \mathsf{N}_t, \mathsf{A}_t, R_g, \inf\{t \geq e \ : \ pred(t)\})
\end{aligned}$$

where $pred(t)$ is simply the state guard predicate where each component count expression $H : P$ for $(H, P) \in \mathcal{B}(G)$, is replaced with the corresponding sum of potentially probed component counts at time t, $\sum_{Q \in P \bowtie_*} \mathsf{N}_t(H, Q)$. Finally:

$$\mathcal{U}(\tilde{G}, \mathsf{N}_t, \mathsf{A}_t, R_g^a[n], e) := \inf\{t \geq e \ : \ \mathcal{U}'(\tilde{G}, \mathsf{N}_t, \mathsf{A}_t, R_g^a, e, t) \geq n\}$$

for $\mathcal{U}'(\tilde{G}, \mathsf{N}_t, \mathsf{A}_t, a_1 \mid \ldots \mid a_k, e, t) := \sum_{i=1}^k \mathsf{A}_t(a_i) - \sum_{i=1}^k \mathsf{A}_e(a_i)$.

As an example, the global probe:

$$\epsilon : \mathsf{start}, (a[10] \mid b[20]), (c[5]; d[30]) : \mathsf{stop}$$

would be translated to the following expression:

$$\max(\inf\{t \geq t_1 \ : \ A_c(t) - A_c(t_1) \geq 5\}, \inf\{t \geq t_1 \ : \ A_d(t) - A_d(t_1) \geq 30\})$$

where:

$$t_1 := \min(\inf\{t \geq 0 \ : \ A_a(t) - A_a(0) \geq 10\}, \inf\{t \geq 0 \ : \ A_b(t) - A_b(0) \geq 20\})$$

The point-mass approximation to the passage time can then be computed by replacing the above components counts with their ODE approximation given by $\mathsf{V}_t \in \mathcal{B}(\tilde{G}) \to \mathbb{R}_+$ and the action counts with their ODE approximation given by $\mathsf{W}_t \in \mathcal{A} \to \mathbb{R}_+$ and then evaluating the resulting expression. That is, evaluating the quantity $\mathcal{U}(\tilde{G}, \mathsf{V}_t, \mathsf{W}_t, R_g, 0)$.

Again, the approximation of a probe-specified global passage-time measurement of the form of Equation 5.22 as described above could be automated. As for the other kinds of passage-time measures discussed in this chapter, support for their direct evaluation is planned for the GPA tool [229, 235].

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

Figure 5.7: Wireless client fluid component in the distributed wireless network model.

5.7 Worked example: massively-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. A client is out of action until its battery has recharged. The wireless clients send data and control 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 will show that, despite the mass parallelism of the model and the many local states in the client state space, we can model and analyse 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 initialises 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 fluid component used to model the wireless client is given in Figure 5.7.

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 *client_shutdown* signal. Alternatively the battery may be recharged a level at any point. The fluid component used to model the battery is given in Figure 5.8.

We then compose the $\boldsymbol{ClientHibernate}$ and $\boldsymbol{Battery}_{n_b}$ components together to form a fluid 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 *client_charge*-action. Once this has happened the *client_on*-action is available and the client can come into operation

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

Figure 5.8: Battery fluid component in the distributed wireless network model.

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

Figure 5.9: Network channel fluid component in the distributed wireless network model.

for another charge cycle. The action set:

$$L_1 := \{client_on, client_off, client_shutdown\}$$

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 modelled using the fluid component given in Figure 5.9.

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:

$$\boldsymbol{DWN}(n_c, n_h) \stackrel{def}{=} \mathbf{Clients}\{\boldsymbol{CB}[n_c]\} \underset{M}{\bowtie} \mathbf{Network}\{\boldsymbol{Channel}[n_h]\}$$

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 chapter. The full systems of first- and second-moment ODEs for the model $\boldsymbol{DWN}(n_c, n_h)$ according to Definition 3.3.1 are available online [131].

5.7.1 Example measurements and results

In this section we will 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 ODE-based 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% confidence. Confidence intervals were computed exactly using *binomial proportion confidence intervals* [194].

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 of the form of Equation 5.14 which expresses this is:

$$
\begin{aligned}
\boldsymbol{PM} \stackrel{def}{=} \ & \mathsf{begin} : \mathsf{start}, \mathsf{end} : \mathsf{stop}^{\hookrightarrow} \\
& \mathsf{observes} \quad \boldsymbol{Probe}_l \stackrel{def}{=} (\mathit{data_tfr}; \mathit{cont_tfr}) \backslash \mathit{client_shutdown} : \mathsf{begin}, \\
& \qquad\qquad \mathit{client_shutdown} : \mathsf{end}^{\hookrightarrow} \\
& \mathsf{where} \quad \mathbf{Clients}\{\boldsymbol{CB}[?n]\} \Longrightarrow \mathbf{Clients}\{(\boldsymbol{CB} \underset{*}{\bowtie} \boldsymbol{Probe}_l) \wr\wr \boldsymbol{CB}[?n-1]\} \\
& \mathsf{in} \quad \boldsymbol{DWN}(n_c, n_h)
\end{aligned}
\tag{5.24}
$$

In order to perform the ODE approximation, we follow Section 5.5.1 and construct the probed iGPEPA models $\widetilde{\boldsymbol{DWN}}(n_c, n_h)$ and $\widetilde{\boldsymbol{DWN}}'(n_c, n_h)$ as follows:

$$
\widetilde{\boldsymbol{DWN}}(n_c, n_h) \stackrel{def}{=} \mathbf{Clients}\{(\boldsymbol{CB} \underset{K}{\bowtie} \boldsymbol{Pb}_S) \wr\wr \boldsymbol{CB}[n_c - 1]\} \underset{M}{\bowtie} \mathbf{Network}\{\boldsymbol{Channel}[n_h]\} \tag{5.25}
$$

and:

$$
\widetilde{\boldsymbol{DWN}}'(n_c, n_h) \stackrel{def}{=} \mathbf{Clients}\{(\boldsymbol{CB} \underset{K}{\bowtie} \boldsymbol{Pb}'_S) \wr\wr \boldsymbol{CB}[n_c - 1]\} \underset{M}{\bowtie} \mathbf{Network}\{\boldsymbol{Channel}[n_h]\} \tag{5.26}
$$

where the local probe $\boldsymbol{Probe}_l$ has been translated to the iPEPA component $\boldsymbol{Pb}_S$ given in Figure 5.10 according to Section 5.3.2, and $K := \{\mathit{client_shutdown}, \mathit{data_tfr}, \mathit{cont_tfr}\}$. The component $\boldsymbol{Pb}'_S$ is the translation of the absorbing version of $\boldsymbol{Probe}_l$.

We then proceed by performing vanishing state removal as in Section 5.2.1 to obtain the derived transition systems of the probed fluid components $\boldsymbol{CB} \underset{K}{\bowtie} \boldsymbol{Pb}_S$ and $\boldsymbol{CB} \underset{K}{\bowtie} \boldsymbol{Pb}'_S$.

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

Figure 5.10: Translation of the local probe in the measure specification of Equation 5.24 to an iPEPA component.

The next stage of the analysis as described in Section 5.5.1 requires us to derive the system of 56 coupled ODEs corresponding to the probed iGPEPA model of Equation 5.25. These are then solved to obtain an approximation to the stationary component-count expectations, say $\tilde{v}_{Y,Q}$ for $(Y,Q) \in \mathcal{B}(\widetilde{\boldsymbol{DWN}}(n_c, n_h))$. This is possible by numerical integration for a sufficiently long period of time, or, often much more efficiently, algebraically to find the 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 Equation 5.26. 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 $\sum_{C \in -\underset{K}{\bowtie} \boldsymbol{Pb}'_{\mathrm{Acc}}} \tilde{v}'_{H,C}(t)$, where $\boldsymbol{Pb}_{\mathrm{Acc}}$ is the accepting state of the local probe $\boldsymbol{Pb}'_S$, and the ODEs are initialised suitably with the quantities $\tilde{v}_{Y,Q}$ computed above as specified by the formula of Equation 5.17. Figure 5.11 compares the actual CDF for increasing numbers of fluid components with the ODE approximation. As expected, we see that the accuracy of the approximation increases as the component populations are scaled up (see Section 6.4.2 for general theoretical justification of this fact).

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 of the form of Equation 5.18 which expresses this is:

$$\begin{aligned}
\boldsymbol{PM} \stackrel{def}{=}\ & \mathsf{begin} : \mathsf{start}, \mathsf{end} : \mathsf{stop} \\
& \mathsf{observes} \quad \boldsymbol{Probe}_l \stackrel{def}{=} cont_tfr : \mathsf{begin}, data_tfr[4] : \mathsf{end} \\
& \mathsf{where} \quad \mathbf{Clients}\{\boldsymbol{CB}[?n]\} \Longrightarrow \mathbf{Clients}\{(\boldsymbol{CB} \underset{*}{\bowtie} \boldsymbol{Probe}_l) \wr\wr \boldsymbol{CB}[?n-1]\} \\
& \mathsf{in} \quad \boldsymbol{DWN}(n_c, n_h)
\end{aligned} \tag{5.27}$$

Note that due to the global probe being non repeating, this passage-time measurement will be 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.

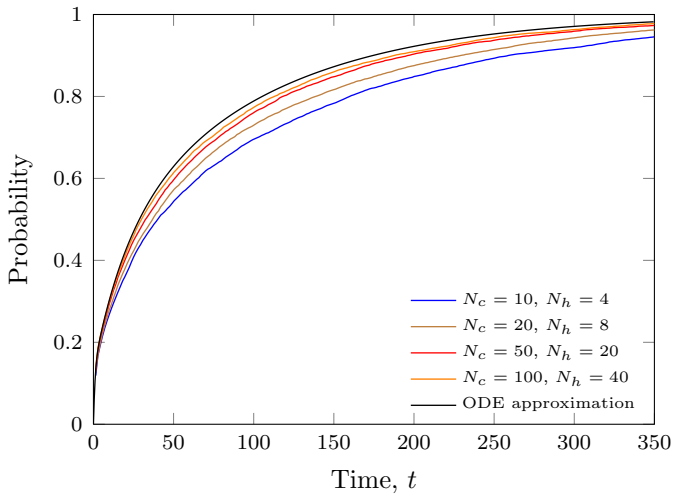


Figure 5.11: Passage-time CDFs for the steady-state individual passage time specified by Equation 5.24 compared with the ODE-derived approximation. Rates: $r_{on} = 0.3$, $r_{off} = 0.6$, $r_{init} = 0.5$, $r_{cont} = 0.85$, $r_{radio} = 0.16$, $r_{charge} = 0.02$, $r_{shutdown} = 1.5$ and $r_{timeout} = 0.2$; and weights: $\omega_{init} = 0.07$ and $\omega_{tfr} = 0.15$.

In order to perform the ODE approximation, we follow Section 5.5.2 and construct the probed iGPEPA model $\widetilde{\boldsymbol{DWN}}(n_c, n_h)$ as follows:

$$\widetilde{\boldsymbol{DWN}}(n_c, n_h) \stackrel{def}{=} \mathbf{Clients}\{(\boldsymbol{CB} \underset{K}{\bowtie} \boldsymbol{Pb}_S) \wr \boldsymbol{CB}[n_c - 1]\} \underset{M}{\bowtie} \mathbf{Network}\{\boldsymbol{Channel}[n_h]\} \quad (5.28)$$

where the local probe $\boldsymbol{Probe}_l$ has been translated to the iPEPA component $\boldsymbol{Pb}_S$ given in Figure 5.12 according to Section 5.3.2, and $K := \{data_tfr, cont_tfr\}$.

We then proceed by performing vanishing state removal as in Section 5.2.1 to obtain the derived transition system of the probed fluid 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 Equation 5.28. The CDF ODE approximation is then given by Equation 5.20 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 5.5.2.

Figure 5.13 compares the actual CDF for increasing numbers of fluid components with the ODE approximation. As expected, we see that the accuracy of the approximation increases as the component populations are scaled up.

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 would be ϵ : start, $client_shutdown[n_c/2]$: stop. However, this would not specify the correct quantity since if a single individual performs $client_shutdown$ more than once, the global probe should only count it the first time. The situation is easily resolved however by attaching an appropriate local probe. A complete measure specification of

$$
\begin{aligned}
\boldsymbol{Pb}_S &\stackrel{def}{=} (cont_tfr, \top).\boldsymbol{Pb}_{Sc} + (data_tfr, \top).\boldsymbol{Pb}_S \\
\boldsymbol{Pb}_{Sc} &\stackrel{def}{=} \mathsf{begin}.\boldsymbol{Pb}_F \\
\boldsymbol{Pb}_F &\stackrel{def}{=} (data_tfr, \top).\boldsymbol{Pb}_{F1d} + (cont_tfr, \top).\boldsymbol{Pb}_F \\
\boldsymbol{Pb}_{F1d} &\stackrel{def}{=} (data_tfr, \top).\boldsymbol{Pb}_{F2d} + (cont_tfr, \top).\boldsymbol{Pb}_{F1d} \\
\boldsymbol{Pb}_{F2d} &\stackrel{def}{=} (data_tfr, \top).\boldsymbol{Pb}_{F3d} + (cont_tfr, \top).\boldsymbol{Pb}_{F2d} \\
\boldsymbol{Pb}_{F3d} &\stackrel{def}{=} (data_tfr, \top).\boldsymbol{Pb}_{F4d} + (cont_tfr, \top).\boldsymbol{Pb}_{F3d} \\
\boldsymbol{Pb}_{F4d} &\stackrel{def}{=} \mathsf{end}.\boldsymbol{Pb}_{\mathrm{Acc}} \\
\boldsymbol{Pb}_{\mathrm{Acc}} &\stackrel{def}{=} (data_tfr, \top).\boldsymbol{Pb}_{\mathrm{Acc}} + (cont_tfr, \top).\boldsymbol{Pb}_{\mathrm{Acc}}
\end{aligned}
$$

Figure 5.12: Translation of the local probe in the measure specification of Equation 5.27 to an *i*PEPA component.

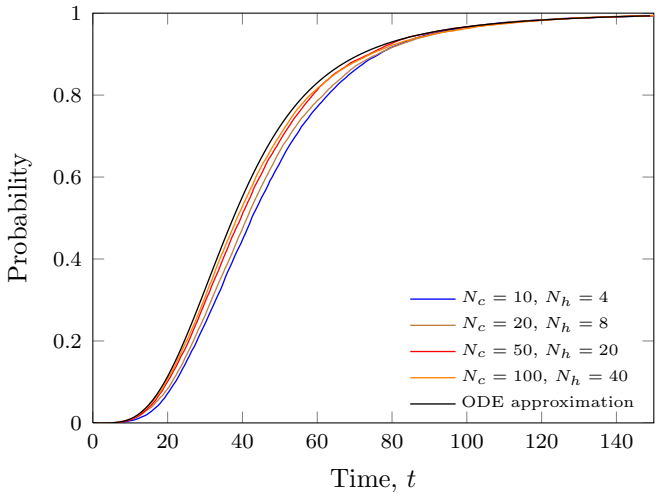


Figure 5.13: Passage-time CDFs for the transient individual passage time specified by Equation 5.27 computed by stochastic simulation compared with the ODE-derived approximation. Rates: $r_{on} = 0.3$, $r_{off} = 0.6$, $r_{init} = 0.5$, $r_{cont} = 0.85$, $r_{radio} = 0.35$, $r_{charge} = 0.05$, $r_{shutdown} = 1.5$ and $r_{timeout} = 0.2$; and weights: $\omega_{init} = 0.07$ and $\omega_{tfr} = 0.15$.

the form of Equation 5.22 which expresses this passage-time quantity is:

$$\begin{aligned} \boldsymbol{PM} \stackrel{def}{=} \epsilon : \mathsf{start}, \mathsf{end}[n_c/2] : \mathsf{stop} \\ \mathsf{observes} \quad \boldsymbol{Probe}_l \stackrel{def}{=} \mathit{client_shutdown} : \mathsf{end} \\ \mathsf{where} \quad \mathbf{Clients}\{\boldsymbol{CB}[?n]\} \Longrightarrow \mathbf{Clients}\{(\boldsymbol{CB} \underset{*}{\bowtie} \boldsymbol{Probe}_l)[?n]\} \\ \mathsf{in} \quad \boldsymbol{DWN}(n_c, n_h) \end{aligned} \tag{5.29}$$

This is now correct because the local probe is non-repeating 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 *i*GPEPA model's derived transition system and the associated system of 26 ODEs — one each whose solution counts the number of fluid components in each of their non-vanishing derivative states, and one, whose solution, $w_{\mathsf{end}}(t)$, corresponds to the end-counting process, $A_{\mathsf{end}}(t)$. In line with Section 5.6, the deterministic approximation to the global passage time of interest is then given by:

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

Figure 5.14 compares the actual CDF for increasing numbers of fluid components with the deterministic ODE approximation and with an example of approximate CDF bounds derived using the second-order Chebyshev's inequality following the method in Section 4.3.2. In order to produce the second-order moment bounds, 350 ODEs must be constructed from Definition 5.4.4. As expected, we see that the accuracy of the deterministic approximation increases as the component populations are scaled up (see Section 6.4.1 for general theoretical justification of this fact). The usefulness of the CDF bound is of course for population sizes for which the deterministic approximation is too coarse, such as the case depicted in Figure 5.14.

5.8 Conclusion

In this chapter, we have introduced the *unified stochastic probe* framework for efficiently and compositionally 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 behaviour 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 modeller and for easy comparison with other techniques such as asCSL [15], CSL$^{\mathrm{TA}}$ [95] and tools, such as PRISM [173] and DNAmaca [164].

In order to support immediate communication between probes in the framework, we have formally extended the stochastic process algebra GPEPA with immediate transitions in a manner which allows the differential-equation approximation techniques of Chapter 3 to remain available. We have also extended the differential-equation analysis to include approximation of action-counting processes which significantly extends the type of global passage-time measurement which can be analysed using differential-equation based techniques.

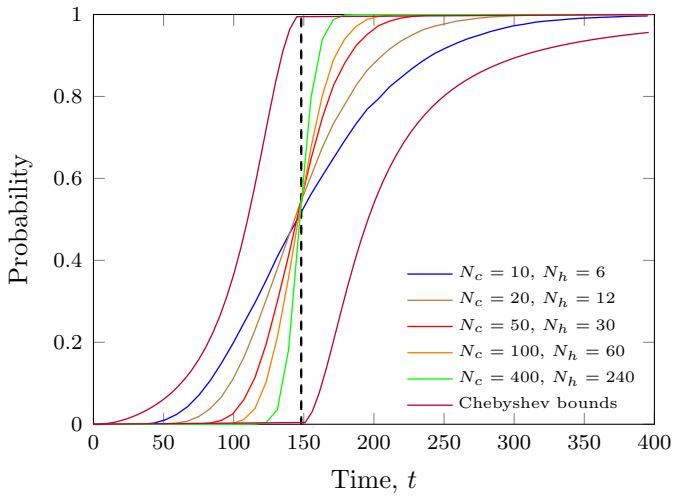


Figure 5.14: Passage-time CDFs for the global passage time specified by Equation 5.29 compared with the ODE-derived point-mass approximation (the dashed line). Also shown are ODE-derived approximate CDF bounds for the case $N_c = 20$, $N_h = 12$. Rates: $r_{on} = 0.3$, $r_{off} = 0.6$, $r_{init} = 0.5$, $r_{cont} = 0.85$, $r_{radio} = 0.4$, $r_{charge} = 0.1$, $r_{shutdown} = 1.5$ and $r_{timeout} = 0.2$; and weights: $\omega_{init} = 0.07$ and $\omega_{tfr} = 0.15$.

Finally, we have identified a large subset of probe-specified measures which are also amenable to efficient differential-equation based analysis. More specifically, *any* steady-state or transient individual passage-time measure (that is, a measure of the form of Equation 5.14 or Equation 5.18, respectively) can be analysed using the ODE approach. Here, there are no restrictions on the form of the local probe specifying the analysis — the entire regular-expression language of Equation 5.4 is available to the modeller. In the case of global passage-time measures, there is again no restriction on the form of the local probes, but they must be applied to all components within a group. Furthermore, the regular expression specifying the corresponding global probe must satisfy a restricted grammar (Equation 5.23). We reiterate that where ODE-based techniques cannot be used to tackle a particular global passage-time probe query, traditional CTMC analysis techniques can be deployed if the model is not so large as to make this infeasible.

Finally, we have demonstrated the combined specification and evaluation techniques on a large and detailed model of a distributed wireless network.

Chapter 6

Theoretical convergence of the first-order approximation

Key contributions
Convergence and bounds in the transient regime
Convergence in the steady-state regime
Mechanistic verification of asymptotic stability for a large class of models

6.1 Introduction

In this chapter, we consider the first-order differential-equation approximations to the counting processes in an *i*GPEPA model as the total population of fluid components is scaled up. We derive bounds on the approximation error in the transient regime and limiting convergence results for both the transient and steady-state regimes. Such results are crucial if we are to have faith in this analysis approach. Although, as we will see, there is much scope for future research in this area.

We now give a summary of the key contributions of the work presented in this chapter which should also serve as a map of the chapter's structure.

Convergence and bounds in the transient regime We show in Section 6.2 that under a suitable rescaling, the counting stochastic processes converge to the solution of the differential equations in the transient case, that is, over bounded intervals of time. Furthermore, we give explicit bounds on the rate of convergence and identify why these bounds appear to be very conservative in practice, suggesting an avenue for future research.

Convergence in the steady-state regime In Section 6.3, we derive an analogous result for the component-counting ODEs in the steady-state regime. This result is conditioned on the ODEs having an asymptotically stable fixed point (one to which all trajectories converge).

Mechanistic verification of asymptotic stability for a large class of models
The steady-state convergence result is only useful formally if we have an efficient means of validating asymptotic stability for a given model's approximating system of ODEs. Using a combination of reduction techniques with methods from control theory we develop such an approach in Sections 6.3.1 and 6.3.2 which is inexpensive and would be straightforward to automate.

In deriving convergence results for the transient and steady-state regimes of GPEPA and *i*GPEPA, we emphasise further that the differential-equation approach is an approximation of the discrete CTMC semantics. In particular, we do not view it here as an orthogonal interpretation of the stochastic process model semantics.

6.1.1 Structurally-equivalent *i*GPEPA models

In this section we will define exactly what we mean by a *sequence of* i*GPEPA models with increasing total component population*. We will introduce the formal notion of a *structurally-equivalent sequence of* i*GPEPA models*. We will also use this section to introduce notation and definitions which will be used throughout the rest of the chapter.

When two *i*GPEPA models are structurally the same, differing only in that they may have different component population sizes, but in the same ratios, we say that they are *structurally equivalent*. A formal definition follows.

Definition 6.1.1 (*i*GPEPA:: Structural equivalence)**.** *Let G_1 and G_2 be two* i*GPEPA models. Then we say they are structurally equivalent if firstly they have the same model structure, $\mathcal{B}(G_1) = \mathcal{B}(G_2) =: \mathcal{B}$ and $\mathcal{W}(G_1, G_2) =$* **true***, where $\mathcal{W}(\cdot,\cdot)$ is defined by:*

$$\mathcal{W}(M_1 \underset{L}{\bowtie} M_2, N_1 \underset{L}{\bowtie} N_2) := \mathcal{W}(M_1, N_1) \wedge \mathcal{W}(M_2, N_2)$$
$$\mathcal{W}(Y\{D_1\}, Y\{D_2\}) := \textbf{true}$$

and **false** *in all other cases. Secondly, they must have the same initial fluid component population ratios, that is, for all $(H, P) \in \mathcal{B}$, $\mathcal{C}(G_1, H, P)/\mathcal{S}(G_1) = \mathcal{C}(G_2, H, P)/\mathcal{S}(G_2)$.*

Following on, a set or sequence of *i*GPEPA models is structurally equivalent, if each pair of models within the set or sequence is structurally equivalent. We have already considered a few such sequences, one example is $\{\boldsymbol{CS}(2n, n)\}_{n=1}^{\infty}$ in terms of the client–server model given in Section 3.2.3. We will usually be interested in such sequences which also have the property that $\mathcal{S}(G^{(i)}) \to \infty$ as $i \to \infty$, that is, the total component population size increases without bound. The example $\{\boldsymbol{CS}(2n, n)\}_{n=1}^{\infty}$ also has this property.

For a given sequence of structurally-equivalent *i*GPEPA models $\{G^{(i)}\}_{i=1}^{\infty}$ we will write, for example, $\mathcal{N} := \mathcal{N}(G^{(i)})$, $\mathcal{B} := \mathcal{B}(G^{(i)})$, $\mathcal{A} := \mathcal{A}(G^{(i)})$ and $\mathcal{J} := \mathcal{J}(G^{(i)})$ for any i, since these quantities are always well defined by structural equivalence. Fixing some ordering for the elements of $\mathcal{B}$ and $\mathcal{A}$, we write also $\mathbf{N}^{(i)}(t) \in \mathbb{Z}_+^{\mathcal{N}}$ for the component-counting stochastic process of $G^{(i)}$, $\mathbf{A}^{(i)}(t) \in \mathbb{Z}_+^{|\mathcal{A}|}$ for the action-counting process, $\mathbf{v}^{(i)}(t) \in \mathbb{R}_+^{\mathcal{N}}$ for the differential-equation approximation to the component counts (Definition 3.3.1) and $\mathbf{w}^{(i)}(t) \in \mathbb{R}_+^{|\mathcal{A}|}$ for the differential-equation approximation to the action counts (Definition 5.4.4).

By structural equivalence, the differential equation approximations for each i satisfy the same ODEs: $\dot{\mathbf{v}}^{(i)}(t) = \mathbf{f}(\mathbf{v}^{(i)}(t))$ for the component counts and $\dot{\mathbf{w}}^{(i)}(t) = \mathbf{g}(\mathbf{v}^{(i)}(t))$ for the action counts.[1] The convergence results will be concerned primarily with the rescaled quantities:

$$\bar{\mathbf{N}}^{(i)}(t) := \mathbf{N}^{(i)}(t)/\mathcal{S}(G^{(i)}) \qquad \bar{\mathbf{A}}^{(i)}(t) := \mathbf{A}^{(i)}(t)/\mathcal{S}(G^{(i)})$$
$$\bar{\mathbf{v}}^{(i)}(t) := \mathbf{v}^{(i)}(t)/\mathcal{S}(G^{(i)}) \qquad \bar{\mathbf{w}}^{(i)}(t) := \mathbf{w}^{(i)}(t)/\mathcal{S}(G^{(i)})$$

Again by homogeneity (Lemma 3.3.2), the rescaled quantities $\bar{\mathbf{v}}^{(i)}(t)$ and $\bar{\mathbf{w}}^{(i)}(t)$ also satisfy the same differential equation and are thus independent of i, so we may write just $\bar{\mathbf{v}}(t)$ and $\bar{\mathbf{w}}(t)$ without ambiguity.

When we wish to specify explicitly a deterministic initial condition for $\mathbf{N}^{(i)}(t)$ we will write $\mathbf{N}^{(i)}_{\mathtt{n}}(t)$ for the process started in state $\mathtt{n} \in \mathbb{Z}_+^{\mathcal{N}}$ corresponding to some aggregate state reachable from the initial state $G^{(i)}$. Similarly, we will also write $\bar{\mathbf{N}}_{\bar{\mathtt{n}}}(t)$ for the rescaled process started in the (rescaled) state $\bar{\mathtt{n}} := \mathtt{n}/\mathcal{S}(G^{(i)})$. We will also use the same subscript notation on ODE solutions to indicate their initial conditions, for example $\bar{\mathbf{v}}_{\bar{\mathtt{n}}}(t)$ for the solution to the initial value problem specified by $\bar{\mathbf{v}}_{\bar{\mathtt{n}}}(0) = \bar{\mathtt{n}}$ and $\dot{\bar{\mathbf{v}}}_{\bar{\mathtt{n}}}(t) = \mathbf{f}(\bar{\mathbf{v}}_{\bar{\mathtt{n}}}(t))$. If we wish to reference elements of the vector $\bar{\mathbf{N}}_{\bar{\mathtt{n}}}(t)$ or $\bar{\mathbf{v}}_{\bar{\mathtt{n}}}(t)$, we will write, say for the kth element, $\bar{N}_{\bar{\mathtt{n}},k}(t)$ or $\bar{v}_{\bar{\mathtt{n}},k}(t)$, respectively.

6.2 Transient bounds and convergence

The following theorem establishes a bound on the probability that an iGPEPA model's rescaled stochastic process will exceed a given maximum deviation from its ODE approximation. It holds transiently, that is, over finite time horizons $[0, T]$. The proof is based on the martingale methodology of Darling and Norris [85] (also see Darling [84]).

Theorem 6.2.1. *Let G be an* i*GPEPA model and assume the notation introduced in Section 6.1.1. Fix $T \geq 0$ and $\epsilon > 0$. Then the following bound holds:*

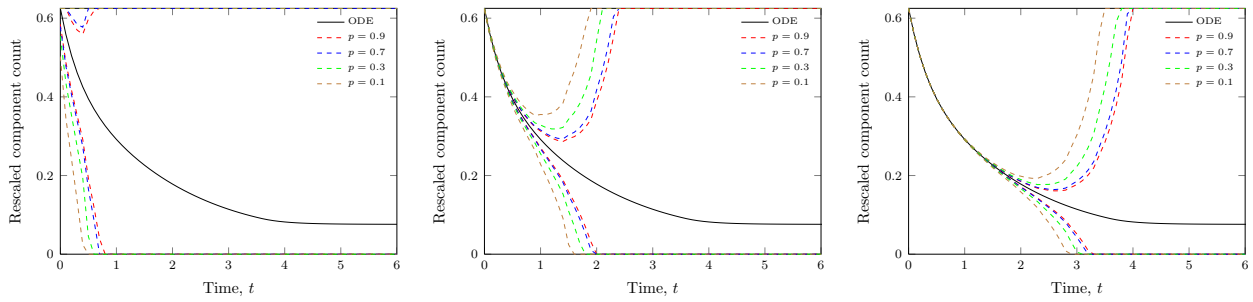
$$\mathbb{P}\left\{\sup_{t\in[0,T]} \|\bar{\mathbf{N}}_{\bar{\mathtt{n}}}(t) - \bar{\mathbf{v}}_{\bar{\mathtt{n}}}(t)\| > \epsilon\right\} \leq \frac{\gamma(\mathcal{S}(G)(\lceil \mathcal{Q}^{max}(G)T\rceil + 1), \mathcal{S}(G)\mathcal{Q}^{max}(G)T)}{(\mathcal{S}(G)(\lceil \mathcal{Q}^{max}(G)T\rceil + 1) - 1)!} + \frac{4(\lceil \mathcal{Q}^{max}(G)T\rceil + 1)\mathcal{N}(G)\exp(2\mathcal{K}(G)T)}{\mathcal{S}(G)\epsilon^2} \tag{6.1}$$

independent of the CTMC initial condition $\bar{\mathtt{n}}$, where $\mathcal{K}(G)$ is a Lipschitz constant of $\mathbf{f}$ such as that given in Lemma A.1.3 and γ is the lower incomplete gamma function.

Proof. See Appendix D.1.1. □

Figure 6.1 shows how this bound performs quantitatively when applied to the client-server example model $\boldsymbol{CS}(n, m)$ of Section 3.2.3. The range of the y-axis coincides with the natural boundaries of the rescaled client population. We see that although the bounds do improve substantially as the component populations are scaled up, they do not behave particularly well with respect to increasing time.

[1] We write $\mathbf{f} : \mathbb{R}_+^{\mathcal{N}} \to \mathbb{R}^{\mathcal{N}}$ and $\mathbf{g} : \mathbb{R}_+^{\mathcal{N}} \to \mathbb{R}^{|\mathcal{A}|}$ for the associated vector fields as given explicitly in Definitions 3.3.1 and 5.4.4 respectively.



(a) $n = 16,000$ (b) $n = 16,000,000$ (c) $n = 16,000,000,000$

Figure 6.1: ϵ-tubes enclosing the rescaled ***Client*** component count around its ODE approximation for a single trace with probability at least $1 - p$ in the model $\boldsymbol{CS}(5n, 3n)$ as computed using Equation 6.1 of Theorem 6.2.1. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 1.0$, $r_{rst} = 1.0$.

The following theorem gives a bound which is usually tighter than that above. Its proof is based on the use of an exponential martingale and is again inspired by Darling and Norris [85]. However, here we provide an explicit optimal expression bounding the error probability, which is not given in Darling and Norris [85]. Exponential martingales of Markov chains have also be used to obtain related bounds in the context of large deviations [221].

Theorem 6.2.2. *Let G be an* iGPEPA *model. Fix $T \geq 0$ and $\epsilon > 0$. Then the following bound holds (with respect to the* maximum norm $\|\mathbf{x}\|_\infty := \max\{|x_1|, \ldots, |x_n|\}$ *for $\mathbf{x} \in \mathbb{R}^n$):*

$$\mathbb{P}\left\{\sup_{t\in[0,T]} \|\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(t) - \bar{\mathbf{v}}_{\bar{\mathbf{n}}}(t)\|_\infty > \epsilon\right\} \leq \frac{2\mathcal{N}(G)\exp\left[\mathcal{S}(G)\epsilon\exp\left(-\sqrt{\mathcal{N}(G)}\mathcal{K}(G)T\right)\right]}{\left(\frac{\epsilon\exp\left(-\sqrt{\mathcal{N}(G)}\mathcal{K}(G)T\right)}{T\mathcal{Q}^{max}(G)} + 1\right)^{\mathcal{S}(G)\left[T\mathcal{Q}^{max}(G)+\epsilon\exp\left(-\sqrt{\mathcal{N}(G)}\mathcal{K}(G)T\right)\right]}} \tag{6.2}$$

independent of the CTMC initial condition $\bar{\mathbf{n}}$. As above, $\mathcal{K}(G)$ is a Lipschitz constant of $\mathbf{f}$ such as that given in Lemma A.1.3.

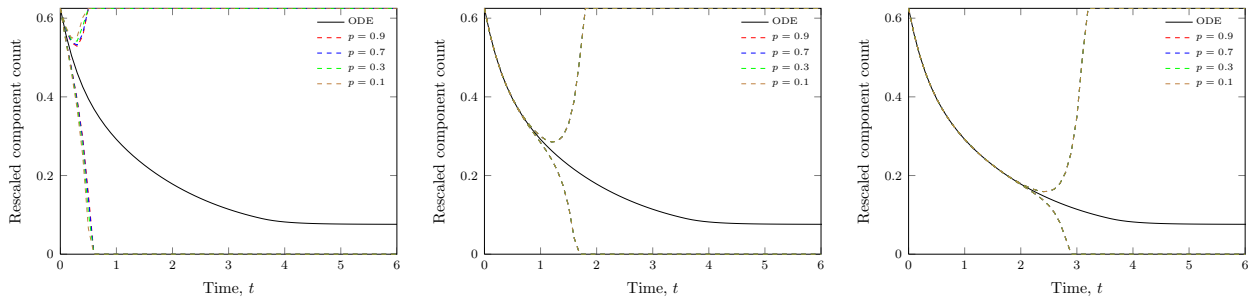
Furthermore, there exists some $D > 0$ independent of $\mathcal{S}(G)$ (specifically, depending on G only through $\mathcal{Q}^{max}(G)$, $\mathcal{K}(G)$ and $\mathcal{N}(G)$), such that:

$$\mathbb{P}\left\{\sup_{t\in[0,T]} \|\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(t) - \bar{\mathbf{v}}_{\bar{\mathbf{n}}}(t)\|_\infty > \epsilon\right\} \leq 2\mathcal{N}(G)\exp(-D\mathcal{S}(G)) \tag{6.3}$$

Proof. See Appendix D.1.2. □

Figure 6.2 shows how the bound of Equation 6.2 performs. Note that in Figures 6.2b and 6.2c, the bounds for different values of p effectively overlay each other. We see that although this represents an improvement over the first bound, it still does not perform particularly well with respect to increasing time, that is, very large component populations are required to obtain a useful bound for any significant interval of time.

In spite of the behaviour of either bound with respect to time, we can use Equation 6.3 to prove limiting convergence in probability to the approximating ODE over bounded intervals of



(a) $n = 16,000$ (b) $n = 16,000,000$ (c) $n = 16,000,000,000$

Figure 6.2: ϵ-tubes enclosing the rescaled ***Client*** component count around its ODE approximation for a single trace with probability at least $1 - p$ in the model $\boldsymbol{CS}(5n, 3n)$ as computed using Equation 6.2 of Theorem 6.2.2. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 1.0$, $r_{rst} = 1.0$.

time as given in the following corollary for both the component counts and the action counts. In preparation for later passage-time convergence results (Section 6.4), we also allow a general initial probability distribution.

Corollary 6.2.3. *Let* $\{G^{(i)}\}_{i=1}^{\infty}$ *be a sequence of structurally-equivalent* iGPEPA *models where* $\mathcal{S}(G^{(i)}) \to \infty$ *as* $i \to \infty$. *Fix* $T \geq 0$ *and* $\epsilon > 0$. *Then as* $i \to \infty$:

$$\mathbb{P}\left\{\sup_{t\in[0,T]} \|\bar{\mathbf{N}}^{(i)}_{\bar{\mathbf{n}}}(t) - \bar{\mathbf{v}}_{\bar{\mathbf{n}}}(t)\| > \epsilon\right\} \longrightarrow 0 \tag{6.4}$$

and:

$$\mathbb{P}\left\{\sup_{t\in[0,T]} \|\bar{\mathbf{A}}^{(i)}_{\bar{\mathbf{n}}}(t) - \bar{\mathbf{w}}_{\bar{\mathbf{n}}}(t)\| > \epsilon\right\} \longrightarrow 0 \tag{6.5}$$

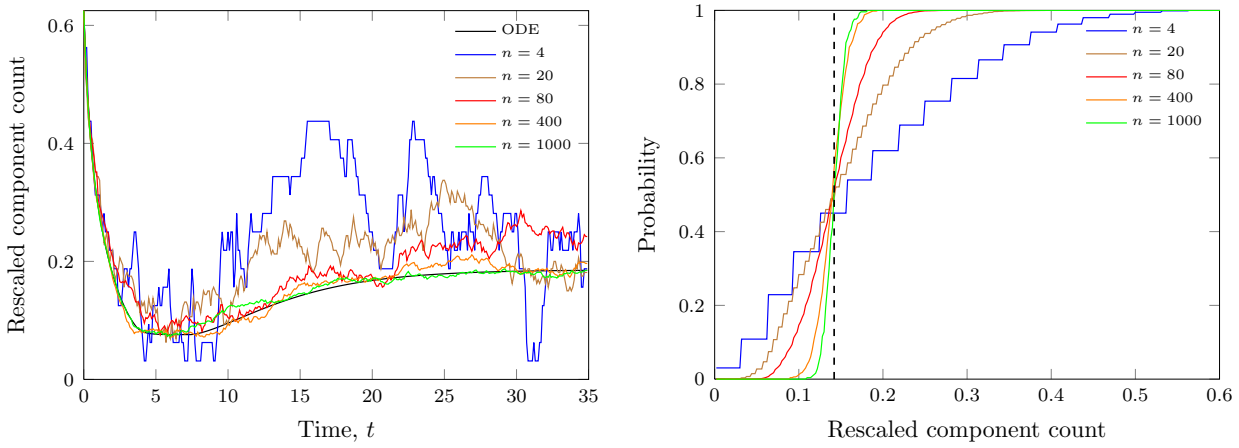
uniformly for all initial states $\bar{\mathbf{n}}$. *Alternatively, if for given* $\bar{\mathbf{n}}$, $\mathbb{P}\left\{\|\bar{\mathbf{N}}^{(i)}(0) - \bar{\mathbf{n}}\| > \delta\right\} \longrightarrow 0$ *as* $i \to \infty$ *for all* $\delta > 0$, *then we have that as* $i \to \infty$, $\mathbb{P}\left\{\sup_{t\in[0,T]} \|\bar{\mathbf{N}}^{(i)}(t) - \bar{\mathbf{v}}_{\bar{\mathbf{n}}}(t)\| > \epsilon\right\} \longrightarrow 0$.

Proof. See Appendix D.1.3. □

Figure 6.3 illustrates this result — single traces of rescaled component counts will eventually overlay the ODE approximation as the population size is increased. We observe that the bounds given by Theorems 6.2.1 and 6.2.2 are generally very loose.

The limit of Equation 6.4 was first given for GPEPA by Hayden and Bradley [140] and for PEPA by Geisweiller et al. [108]. The work of Geisweiller et al. [108] and since then, others [91, 249, 250] gave convergence in probability limit theorems for PEPA models derived using the results of Kurtz [166, 167, 169] (see also Ethier and Kurtz [97]) usually phrased in terms of *density-dependent Markov chains*. These results provide an alternative way to obtain the kind of convergence statements in Corollary 6.2.3, however explicit bounds on the rate of the convergence (as given here in Theorems 6.2.1 and 6.2.2) are not immediate from Kurtz's work.

In the next section, we present results which extend those of Corollary 6.2.3 to also hold as $t \to \infty$, that is, in the steady-state or stationary regime. Since the bounds of Theorems 6.2.1



(a) Single traces

(b) CDF comparison at $t = 15$ (dashed line is ODE point-mass approximation)

Figure 6.3: Comparison of ***Client*** component count simulation results with ODE approximation for the model $\boldsymbol{CS}(5n, 3n)$. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.27$, $r_b = 0.1$, $r_d = 1.0$, $r_{rst} = 1.0$.

and 6.2.2 explode as $t \to \infty$, we cannot hope to extend their proofs to the steady-state case in a straightforward manner.

In fact, the reason for this behaviour is the judicious use of an inequality due to Grönwall in the proofs of this section. The inequality says that for any real-valued integrable function g on $[0, S]$, $g(s) \leq C + D \int_0^s g(u)\,\mathrm{d}u$ for all $s \in [0, S]$ implies $g(S) \leq C \exp(DS)$ (see e.g. Ethier and Kurtz [97, page 498]). This is very much a worst-case device which assumes essentially that perturbed trajectories of the approximating system of ODEs separate at the fastest theoretically-possible rate, which is exponentially with time.

In Figure 6.3, we can see that for models containing only thousands of components, we have impressive empirical convergence. This suggests that convergence to the ODE solution should occur much faster than the rate given in Theorems 6.2.1 and 6.2.2 would suggest. In particular, we predict that these improvements could be found by more careful consideration of the stability of the ODE trajectories for a given model. Similarly, in order to say anything about the $t \to \infty$ case, it would seem necessary to impose conditions on the long-term stability of solutions to the approximating system of ODEs. Indeed, in the next section, we see that under the assumption of an asymptotically stable fixed point, we may prove a steady-state analogue of Corollary 6.2.3. We will also present an inexpensive method for verifying the required stability properties for a large class of *i*GPEPA models.

6.3 Steady-state convergence

In this section, we give a version of Corollary 6.2.3 which is valid as $t \to \infty$, that is, in the steady-state or stationary regime. This result will have a particularly useful form if we are first able to guarantee asymptotic stability of the fixed point of the ODEs approximating the component counts. That is, we would like to be able to guarantee that a trajectory starting from any initial condition converges to the fixed point in the limit as $t \to \infty$. Empirically, this

appears to be the case more often than not but can be difficult to guarantee formally. For this reason, the following two sections are devoted to developing an inexpensive method which will allow formal verification of asymptotic stability of the approximating ODEs for a large class of *i*GPEPA model.

Specifically, in the next section we will see why the system of ODEs given in Definition 3.3.1 cannot in general be expected to have a unique fixed point and thus no fixed point will be *globally* asymptotically stable. We will then show how this issue can potentially be addressed by instead considering asymptotic stability only on a restricted space and global asymptotic stability of a reduced system of ODEs. Then in Section 6.3.2 we provide an efficient technique for the formal validation of this global asymptotic stability for a large class of models. Finally in Section 6.3.3, we give the steady-state convergence theorem.

6.3.1 Asymptotic stability on the reachable subset and the reduced ODE

In order for a system of ODEs to have a globally asymptotically stable fixed point, it must at least have a *unique* fixed point. For the system $\dot{\mathbf{v}}(t) = \mathbf{f}(\mathbf{v}(t))$, we should never expect this to be the case when considered over all of $\mathbb{R}^{\mathcal{N}}$. To see why, we can consider again the client–server model $\boldsymbol{CS}(n, m)$ and recall that the system of ODEs approximating the component counts is the same for any value of n and m. However, we would not expect the long-term behaviour of the ODE also to be independent of n and m. For example, consider a system with $n = 100$ clients and the difference between the long-term evolution with $m = 50$ servers and $m = 20$ servers. Indeed, a fixed point of the approximating ODEs (Equation 3.4) corresponding to the former case is:[2]

$$(25.28, 12.45, 62.27, 8.09, 37.36, 4.55)^T$$

and to the latter case is:

$$(70.11, 4.98, 24.91, 3.24, 14.94, 1.82)^T$$

In order to handle this, associated to an *i*GPEPA model G, we will define a subset of $\mathbb{R}^{\mathcal{N}(G)}$ which contains the reachable subset of the aggregated state space of G and talk instead of asymptotic stability of an ODE fixed point restricted to that subset. That is, we will be interested in verifying only that all trajectories starting in said subset converge to the fixed point. No such condition will be placed on trajectories starting outside of the subset.

To proceed, we decompose the function $\mathbf{f}(\mathbf{x}) = \mathbf{S}\mathbf{r}(\mathbf{x})$ into the product of a matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{N}(G)|\times|\mathcal{J}(G)|}$ and a vector-valued function $\mathbf{r} : \mathbb{R}_+^{|\mathcal{N}(G)|} \to \mathbb{R}_+^{|\mathcal{J}(G)|}$. Each column of $\mathbf{S}$ corresponds to an entry of $\mathcal{J}(G)$ and thus a possible way in which fluid component derivative states may evolve in synchronisation as part of one timed transition. In order to define $\mathbf{S}$ formally we must fix an ordering on the elements of $\mathcal{J}(G)$, so write its jth element as (J_-^j, J_+^j, α_j). Then the ij-entry of $\mathbf{S}$ is defined as 1 if the ith fluid component derivative state in $\mathcal{B}(G)$ (under the same ordering as in Section 6.1.1) is in $J_+^j \setminus J_-^j$, -1 if it is in $J_-^j \setminus J_+^j$ and 0 otherwise. Therefore each column specifies the change in each fluid component derivative state count when the corresponding synchronisation occurs. In models of chemical reactions, the analogous matrix is often called the *stoichiometric matrix* [217] and has previously been referred to as the *activity matrix* by Hillston [147] and Ding [91] in the context of PEPA. For consistency we will also

[2]With rate parameter values: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.2$, $r_b = 0.1$, $r_d = 1.0$, $r_{rst} = 1.0$.

adopt the latter term for iGPEPA models. We refer to the function $\mathbf{r}(\mathbf{x})$ as the *rate vector* and the jth element of this vector is the quantity $\rho_\alpha(J_-^j, J_+^j)\mathcal{R}_\alpha(G, \mathbf{x}, J_-^j)$, which is the rate at which such a transition occurs (Theorem 3.4.3). In the case of the client–server model $\boldsymbol{CS}(n, m)$ of Section 3.2.3, these quantities are:[3]

$$\mathbf{S} = \begin{pmatrix} -1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & -1 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 \end{pmatrix} \qquad \mathbf{r}(\mathbf{x}) = \begin{pmatrix} \min(x_1, x_4) r_r \\ \min(x_2, x_5) r_d \\ x_2 r_{tmt} \\ x_3 r_t \\ x_4 r_b \\ x_5 r_b \\ x_6 r_{rst} \end{pmatrix} \tag{6.6}$$

Now for some iGPEPA model G, write $\bar{\mathtt{n}}_0 \in \mathbb{R}^{|\mathcal{N}(G)|}$ for the rescaled vector of component counts corresponding to the model's initial state. Then the definition of $\mathbf{S}$ and Theorem 3.4.3 together show that any reachable state in the rescaled aggregated CTMC of G is contained in the following set, which we denote by $\mathcal{R}(G)$:

$$\mathcal{R}(G) := \left\{ \bar{\mathtt{n}}_0 + \mathbf{S}\mathbf{x} \ : \ \mathbf{x} \in \mathbb{R}^{|\mathcal{J}(G)|} \right\} \tag{6.7}$$

We note that this definition is similar to that of the *linearised reachability set* often used in the context of stochastic Petri nets [225]. Here it is constructed to encode the reachable discrete state space efficiently for use in various structural verification tasks. In the case of a structurally-equivalent sequence of iGPEPA models $\{G^{(i)}\}_{i=1}^{\infty}$, $\mathcal{R} := \mathcal{R}(G^{(i)})$ is independent of i and contains the rescaled aggregated state space of all of the models in the sequence. Furthermore for $\bar{\mathtt{r}} = \bar{\mathtt{n}}_0 + \mathbf{S}\mathbf{x} \in \mathcal{R}$, we have:

$$\bar{\mathbf{v}}_{\bar{\mathtt{r}}}(t) = \bar{\mathtt{n}}_0 + \mathbf{S}\mathbf{x} + \int_0^t \mathbf{S}\mathbf{r}(\bar{\mathbf{v}}_{\bar{\mathtt{r}}}(s))\,\mathrm{d}s = \bar{\mathtt{n}}_0 + \mathbf{S}\left(\mathbf{x} + \int_0^t \mathbf{r}(\bar{\mathbf{v}}_{\bar{\mathtt{r}}}(s))\,\mathrm{d}s\right)$$

so that $\bar{\mathbf{v}}_{\bar{\mathtt{r}}}(t) \in \mathcal{R}$ for all $t \in \mathbb{R}_+$.

In many cases, the ODEs associated to a structurally-equivalent sequence will have a unique fixed point within $\mathcal{R}$. In the case of the client–server model with ODEs given by Equation 3.4 and assuming we are interested in the structurally-equivalent sequence $\{\boldsymbol{CS}(2n, n)\}_{n=1}^{\infty}$ (so $\bar{\mathtt{n}}_0 = (2/3, 0, 0, 1/3, 0, 0)^T$), this single fixed point is:

$$\bar{\mathtt{r}}^* = (0.1685, 0.0830, 0.4151, 0.0539, 0.2491, 0.0303)^T \tag{6.8}$$

In order to apply the convergence theorem of the next section, we will usually wish to show that such a fixed point is asymptotically stable within $\mathcal{R}$. That is, that any ODE trajectory starting within $\mathcal{R}$ converges towards the fixed point as $t \to \infty$, or formally, if $\bar{\mathtt{r}} \in \mathcal{R}$, $\lim_{t\to\infty} \bar{\mathbf{v}}_{\bar{\mathtt{r}}}(t) = \bar{\mathtt{r}}^*$. Such a question of asymptotic stability on a subset can be recast as a question of global asymptotic stability on all of $\mathbb{R}^m$ for some m by a suitable reduction of the system of ODEs. Specifically, construct a matrix $\mathbf{R} \in \mathbb{R}^{\mathcal{N}\times m}$ where m is the rank of $\mathbf{S}$ by choosing a maximal set of linearly-independent columns of $\mathbf{S}$. Then let $\mathbf{B} \in \mathbb{R}^{m\times\mathcal{N}}$ be its *Moore–Penrose pseudoinverse*

[3] Where we order $\mathcal{B}(\boldsymbol{CS}(n, m))$ according to the order of definition of the fluid component derivative states in Section 3.2.3 and $\mathcal{J}(\boldsymbol{CS}(n, m))$ as enumerated in Section 3.4.4.

which is defined and unique for all real matrices [28].[4] Since $\mathbf{R}$ has full column rank it then holds that $\mathbf{BR} = \mathbf{I}_m$ where $\mathbf{I}_m$ is the $m \times m$ identity matrix and, furthermore, the matrix $\mathbf{RB}$ is an *orthogonal projector* onto the range of $\mathbf{R}$ [28], that is, $\mathbf{RB}$ sends vectors in $\mathbb{R}^{\mathcal{N}}$ to their unique orthogonal projection onto the range of $\mathbf{R}$. In particular, since the columns of $\mathbf{S}$ are already in the range of $\mathbf{R}$ by construction, it holds that $\mathbf{RBS} = \mathbf{S}$.

The system of rescaled *reduced component-counting ODEs* corresponding to an *i*GPEPA model (or structurally-equivalent sequence) is then defined as $\dot{\hat{\mathbf{v}}}(t) = \mathbf{BSr}(\mathbf{R}\hat{\mathbf{v}}(t) + \bar{\mathbf{n}}_0)$. We note that $\mathcal{R} = \left\{\bar{\mathbf{n}}_0 + \mathbf{Sx} \ : \ \mathbf{x} \in \mathbb{R}^{|\mathcal{J}(G)|}\right\} = \{\bar{\mathbf{n}}_0 + \mathbf{Rx} \ : \ \mathbf{x} \in \mathbb{R}^m\}$ and let $\bar{\mathbf{r}} = \bar{\mathbf{n}}_0 + \mathbf{Rx} \in \mathcal{R}$ be arbitrary. We will first show that $\hat{\mathbf{v}}_{\mathbf{x}}(t)$ and $\bar{\mathbf{v}}_{\bar{\mathbf{r}}}(t)$ can be recovered from one another in a straightforward manner. In particular, write $\bar{\mathbf{z}}(t) := \mathbf{R}\hat{\mathbf{v}}_{\mathbf{x}}(t) + \bar{\mathbf{n}}_0$. Then we will show that $\bar{\mathbf{z}}(t) = \bar{\mathbf{v}}_{\bar{\mathbf{r}}}(t)$ for all $t \in \mathbb{R}_+$, specifically:

$$\dot{\bar{\mathbf{z}}}(t) = \mathbf{R}\dot{\hat{\mathbf{v}}}_{\mathbf{x}}(t) = \mathbf{RBSr}(\bar{\mathbf{z}}(t)) = \mathbf{Sr}(\bar{\mathbf{z}}(t))$$

and furthermore we have that $\bar{\mathbf{z}}(0) = \bar{\mathbf{r}}$, so then $\bar{\mathbf{z}}(t) = \bar{\mathbf{v}}_{\bar{\mathbf{r}}}(t) = \mathbf{R}\hat{\mathbf{v}}_{\mathbf{x}}(t) + \bar{\mathbf{n}}_0$ for all $t \in \mathbb{R}_+$. From this, it is then immediate that also $\hat{\mathbf{v}}_{\mathbf{x}}(t) = \mathbf{B}(\bar{\mathbf{v}}_{\bar{\mathbf{r}}}(t) - \bar{\mathbf{n}}_0)$ for all $t \in \mathbb{R}_+$.

Now we will show that that $\mathbf{x}^* \in \mathbb{R}^m$ is globally asymptotically stable (over all of $\mathbb{R}^m$) for the reduced ODE if and only if $\bar{\mathbf{r}}^* := \bar{\mathbf{n}}_0 + \mathbf{Rx}^* \in \mathcal{R}$ is asymptotically stable restricted to $\mathcal{R}$ for the original ODE.[5] Start by assuming that $\mathbf{x}^* \in \mathbb{R}^m$ is globally asymptotically stable for the reduced ODE. That is, for all $\mathbf{x} \in \mathbb{R}^m$, we have $\lim_{t\to\infty} \hat{\mathbf{v}}_{\mathbf{x}}(t) = \mathbf{x}^*$. If $\bar{\mathbf{r}} = \bar{\mathbf{n}}_0 + \mathbf{Rx} \in \mathcal{R}$ is arbitrary then it follows immediately that $\lim_{t\to\infty} \bar{\mathbf{v}}_{\bar{\mathbf{r}}}(t) = \bar{\mathbf{n}}_0 + \mathbf{Rx}^*$. Working in the other direction, assume that $\bar{\mathbf{r}}^* = \bar{\mathbf{n}}_0 + \mathbf{Rx}^* \in \mathcal{R}$ is asymptotically stable restricted to $\mathcal{R}$ for the original ODE, so $\lim_{t\to\infty} \bar{\mathbf{v}}_{\bar{\mathbf{r}}}(t) = \bar{\mathbf{r}}^*$ for all $\bar{\mathbf{r}} = \bar{\mathbf{n}}_0 + \mathbf{Rx} \in \mathcal{R}$. The fact that $\lim_{t\to\infty} \hat{\mathbf{v}}_{\mathbf{x}}(t) = \mathbf{x}^*$ for all $\mathbf{x} \in \mathbb{R}^m$ then also follows directly.

In the case of the client–server model, $\text{rank}(\mathbf{S}) = 4$ (recall that for this model $\mathbf{S}$ was given earlier in Equation 6.6) and we may take $\mathbf{R}$ and $\mathbf{B}$ as follows:

$$\mathbf{R} = \begin{pmatrix} -1 & 0 & 1 & 0 \\ 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{B} = \frac{1}{3}\begin{pmatrix} -1 & -1 & 2 & -2 & 1 & 1 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ 1 & -2 & 1 & -2 & 1 & 1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

Then the reduced four-dimensional system of rescaled ODEs is $\dot{\hat{\mathbf{v}}}(t) = \mathbf{BSr}(\mathbf{R}\hat{\mathbf{v}}(t) + \bar{\mathbf{n}}_0)$, which, explicitly, evaluates to:

$$\dot{\hat{\mathbf{v}}}(t) = \begin{pmatrix} \min(\hat{v}_2(t) - \hat{v}_1(t) + 1/3, \hat{v}_3(t) - \hat{v}_1(t) + 2/3)r_r - r_t\hat{v}_2(t) + r_b(\hat{v}_2(t) - \hat{v}_1(t) + 1/3) - r_{rst}\hat{v}_4(t) \\ \min(\hat{v}_1(t) - \hat{v}_2(t) - \hat{v}_3(t), \hat{v}_1(t) - \hat{v}_2(t) - \hat{v}_4(t))r_d - r_t\hat{v}_2(t) \\ r_b(-\hat{v}_1(t) + \hat{v}_2(t) + 1/3) + r_{tmt}(\hat{v}_1(t) - \hat{v}_2(t) - \hat{v}_3(t)) - r_{rst}\hat{v}_4(t) \\ 1/3(r_b(1 - 3\hat{v}_4(t)) - 3r_{rst}\hat{v}_4(t)) \end{pmatrix} \tag{6.9}$$

[4] Since $\mathbf{R}$ has full rank by construction, its Moore–Penrose pseudoinverse can be computed directly as $(\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T$ if $m < \mathcal{N}$ and is simply $\mathbf{R}^{-1}$ if $m = \mathcal{N}$ [28].

[5] Every element of $\mathcal{R}$ has a *unique* decomposition of the form $\bar{\mathbf{n}}_0 + \mathbf{Rx}$ because $\mathbf{R}$ has full column rank.

where we are again considering the structurally-equivalent sequence $\{\boldsymbol{CS}(2n, n)\}_{n=1}^{\infty}$ so have chosen $\bar{\mathbf{n}}_0 = (2/3, 0, 0, 1/3, 0, 0)^T$.

In the next section we present a technique for verifying global asymptotic stability of the reduced ODE in an inexpensive fashion for a large class of *i*GPEPA models. By the above discussion, this will in turn prove the required asymptotic stability of the original ODE when restricted to the set $\mathcal{R}$.

6.3.2 Verification of global asymptotic stability of the reduced ODE

As discussed, we seek in this section to present an inexpensive technique for verifying global asymptotic stability of the reduced ODE for some *i*GPEPA model G (or structurally-equivalent sequence). By the results of the previous section, this will in turn imply asymptotic stability for the original system of component-counting ODEs restricted to the set $\mathcal{R}(G)$.

A common approach to verifying global asymptotic stability is to construct an appropriate *Lyapunov function*. Assume that we wish to verify global asymptotic stability of some differential equation $\dot{\mathbf{y}}(t) = \mathbf{k}(\mathbf{y}(t))$ where $\mathbf{y}(t) \in \mathbb{R}^n$ for $t \in \mathbb{R}_+$ and $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^n$ for the unique fixed point $\mathbf{y}^* \in \mathbb{R}^n$. Then if a function $V : \mathbb{R}^n \to \mathbb{R}$ exists which satisfies:

1. $V(\mathbf{y}^*) = 0$;
2. $V(\mathbf{y}) > 0$ and $\frac{d(V \circ \mathbf{y})}{dt} < 0$ for all $\mathbf{y} \neq \mathbf{y}^*$; and
3. V is *radially unbounded*, i.e. $\|\mathbf{y}\| \to \infty \Rightarrow V(\mathbf{y}) \to \infty$

then it is called a Lyapunov function and $\mathbf{y}^*$ is globally asymptotically stable [e.g. 163, 187].

For general non-linear systems, the verification of stability properties by Lyapunov function based techniques or otherwise is a very difficult task [163]. However, recall the class of *split-free* *i*GPEPA models which was defined in Section 3.3.1 — informally, a split-free model is one where no synchronised action is enabled by more than one fluid component derivative state on one side of a cooperation between component groups. For such models it was shown that the component rate function consists only of minimum functions and linear combinations so that it is a piecewise linear function of the component counts. By the definition of the right-hand side of the component-counting ODEs (Definition 3.3.1), it follows that this is also piecewise linear. Recall that we denote the right-hand side of the original component-counting ODEs by the function $\mathbf{f}(\bar{\mathbf{v}}(t))$. Furthermore, the right-hand side of the reduced system of ODEs as given in the previous section can then be given in terms of $\mathbf{f}$ by $\hat{\mathbf{f}}(\hat{\mathbf{v}}(t)) := \mathbf{B}\mathbf{f}(\mathbf{R}\hat{\mathbf{v}}(t) + \bar{\mathbf{n}}_0)$ and so is itself always piecewise affine. Specifically, we may re-write it in the following form:

$$\dot{\hat{\mathbf{v}}}(t) = \mathbf{A}_i\hat{\mathbf{v}}(t) + \mathbf{b}_i \qquad \text{when } \hat{\mathbf{v}}(t) \in \Gamma_i \text{ for } i = 1, \ldots, d$$

Furthermore, the regions Γ_i corresponding to each different affine dynamic are separated by affine hyperplanes in $\mathbb{R}^m$. Since the right-hand side is continuous, each such region Γ_i can be chosen to also include the segments of the affine hyperplanes which define its boundaries — we assume here that this is the case. For example, the reduced ODEs for the client–server model given

in Equation 6.9 can be written in this form where $d = 4$ and we use $\bar{\mathbf{n}}_0 = (2/3, 0, 0, 1/3, 0, 0)^T$ as before:

$$\mathbf{A}_1 = \begin{pmatrix} -r_b - r_r & -r_t + r_b & r_r & -r_{rst} \\ r_d & -r_d - r_t & -r_d & 0 \\ -r_b + r_{tmt} & r_b - r_{tmt} & -r_{tmt} & -r_{rst} \\ 0 & 0 & 0 & -r_{rst} - r_b \end{pmatrix} \qquad \mathbf{b}_1 = \frac{1}{3}\begin{pmatrix} r_b + 2r_r \\ 0 \\ r_b \\ r_b \end{pmatrix}$$

$$\Gamma_1 = \{\mathbf{x} \in \mathbb{R}^4 \ : \ x_3 - x_2 + 1/3 \leq 0,\ x_4 - x_3 \leq 0\}$$

$$\mathbf{A}_2 = \begin{pmatrix} -r_b - r_r & -r_t + r_b + r_r & 0 & -r_{rst} \\ r_d & -r_d - r_t & -r_d & 0 \\ -r_b + r_{tmt} & r_b - r_{tmt} & -r_{tmt} & -r_{rst} \\ 0 & 0 & 0 & -r_{rst} - r_b \end{pmatrix} \qquad \mathbf{b}_2 = \frac{1}{3}\begin{pmatrix} r_b + r_r \\ 0 \\ r_b \\ r_b \end{pmatrix}$$

$$\Gamma_2 = \{\mathbf{x} \in \mathbb{R}^4 \ : \ x_3 - x_2 + 1/3 \geq 0,\ x_4 - x_3 \leq 0\}$$

$$\mathbf{A}_3 = \begin{pmatrix} -r_b - r_r & -r_t + r_b & r_r & -r_{rst} \\ r_d & -r_d - r_t & 0 & -r_d \\ -r_b + r_{tmt} & r_b - r_{tmt} & -r_{tmt} & -r_{rst} \\ 0 & 0 & 0 & -r_{rst} - r_b \end{pmatrix} \qquad \mathbf{b}_3 = \frac{1}{3}\begin{pmatrix} r_b + 2r_r \\ 0 \\ r_b \\ r_b \end{pmatrix}$$

$$\Gamma_3 = \{\mathbf{x} \in \mathbb{R}^4 \ : \ x_3 - x_2 + 1/3 \leq 0,\ x_4 - x_3 \geq 0\}$$

$$\mathbf{A}_4 = \begin{pmatrix} -r_b - r_r & -r_t + r_b + r_r & 0 & -r_{rst} \\ r_d & -r_d - r_t & 0 & -r_d \\ -r_b + r_{tmt} & r_b - r_{tmt} & -r_{tmt} & -r_{rst} \\ 0 & 0 & 0 & -r_{rst} - r_b \end{pmatrix} \qquad \mathbf{b}_4 = \frac{1}{3}\begin{pmatrix} r_b + r_r \\ 0 \\ r_b \\ r_b \end{pmatrix}$$

$$\Gamma_4 = \{\mathbf{x} \in \mathbb{R}^4 \ : \ x_3 - x_2 + 1/3 \geq 0,\ x_4 - x_3 \geq 0\}$$

Fortunately, for such continuous piecewise affine systems, it is often possible to inexpensively and automatically compute a suitable Lyapunov function. In the field of control theory, one such fairly recent approach is to define a *common quadratic Lyapunov function* by solving a system of *linear matrix inequalities (LMIs)* [200]. Determining feasible solutions to systems of LMIs is essentially a convex optimisation problem [48] and can thus be done very efficiently. We adopt this approach here and the following theorem gives a system of LMIs (Equation 6.10) for which a solution is a sufficient condition for global asymptotic stability of the reduced ODEs' fixed point.

Theorem 6.3.1. *Let G be a split-free* i*GPEPA model. Then by the above discussion the reduced rescaled ODE is piecewise affine, say:*

$$\dot{\hat{\mathbf{v}}}(t) = \hat{\mathbf{f}}(\hat{\mathbf{v}}(t)) = \mathbf{A}_i\hat{\mathbf{v}}(t) + \mathbf{b}_i \qquad \text{when } \hat{\mathbf{v}}(t) \in \Gamma_i \text{ for } i = 1, \ldots, d$$

Furthermore, the regions of different affine dynamics are separated by affine hyperplanes.

Now assume that we can find a single symmetric positive definite matrix $\mathbf{P} \in \mathbb{R}^{m \times m}$ and $\alpha > 0$ such that the following LMI holds simultaneously for all $i = 1, \ldots, d$:[6]

$$\mathbf{P}\mathbf{A}_i + \mathbf{A}_i^T\mathbf{P} + 2\alpha\mathbf{P} < 0 \tag{6.10}$$

Let $\hat{\mathbf{v}}^*$ be a fixed point of the reduced ODEs. Then $\hat{\mathbf{v}}^*$ is globally asymptotically stable.

[6] Inequalities for matrices, e.g. $Q < 0$ or $Q > 0$ are interpreted as statements of negative or positive definiteness, respectively.

Proof. See Appendix D.1.4. □

Where they exist, feasible solutions to the system of LMIs of Equation 6.10 can be found very quickly using, for example, the MATLAB® LMI toolbox [243]. For the client–server model, there are five individual LMIs — Equation 6.10 for each $i = 1, 2, 3, 4$ and also the LMI $\mathbf{P} > 0$ expressing that $\mathbf{P}$ be positive definite.[7] This information can be input to MATLAB® using the `lmiterm` function along with the requirement that $\mathbf{P}$ be symmetric. The `feasp` function can then be used to compute a solution matrix $\mathbf{P}$ if it exists. In the case of the client-server model (using the same parameters as before), the LMIs are shown feasible in under a second on a standard Intel Core 2 Duo desktop computer — one solution is:

$$\mathbf{P} = \begin{pmatrix} 23.8370 & 6.5895 & 0.5508 & -13.9915 \\ 6.5895 & 47.7363 & -30.6028 & 7.3098 \\ 0.5508 & -30.6028 & 214.8347 & -159.9027 \\ -13.9915 & 7.3098 & -159.9027 & 223.5117 \end{pmatrix} \tag{6.11}$$

By Theorem 6.3.1, this verifies that the unique fixed point $(0.6945, 0.4151, 0.1963, 0.0303)^T$ of the rescaled reduced ODEs is indeed globally asymptotically stable. As discussed in Section 6.3.1, this in turn verifies that the unique fixed point of the original system of ODEs given earlier by Equation 6.8 is also asymptotically stable when restricted to $\mathcal{R}(\boldsymbol{CS}(2n, n))$.

It should be noted that the existence of a solution to the system of LMIs of Equation 6.10 is only a sufficient condition for global asymptotic stability. In practice we have found it to be a very powerful technique — we have not yet found any cases where we suspect global asymptotic stability but Equation 6.10 does not hold. Indeed, in Hayden et al. [142] we have applied this approach to a more detailed model of a customer–service system with multiple customer classes and server preemption and failure. This GPEPA model includes 14 fluid components and the corresponding system of 64 LMIs is solved in under a second on a standard desktop Intel Core 2 Duo computer.

Should we require a stronger test in the future, however, a possible direction is the construction of *piecewise quadratic Lyapunov functions* [155], which we do not consider further in this thesis.

To conclude this section we will give an example of an *i*GPEPA model whose ODEs appear to have a more complicated attractive limit cycle consisting of sustained oscillations which continue indefinitely. This confirms that the test given by Theorem 6.3.1 is very necessary in that we cannot always expect asymptotic stability from systems of ODEs derived from *i*GPEPA models. We note here that this example answers a conjecture of Ding [91, Conjecture 1] in the negative that the system of ODEs derived from any PEPA model has a (constant) limit as time tends to infinity.

The *i*GPEPA definition $\boldsymbol{Osc}(n)$ of this counter example is given in Figure 6.4. Following the method of the previous section, we obtain the following reduced system of rescaled ODEs associated to this structurally-equivalent sequence of models:

$$\dot{\hat{v}}_1(t) = \min(10/23 - \hat{v}_1(t), 10/23 - \hat{v}_2(t))r_r - \min(3/46, 10/23 - \hat{v}_1(t))r_s$$
$$\dot{\hat{v}}_2(t) = \min(\hat{v}_1(t), \hat{v}_2(t))r_q - \min(3/46, \hat{v}_2(t))r_z$$

[7] When computing results, we actually specify the stronger condition $\mathbf{P} - \mathbf{I}_m > 0$ since it is known to be more numerically stable than $\mathbf{P} > 0$ [243].

$$\boldsymbol{P}_1 \stackrel{def}{=} (a, r_r).\boldsymbol{P}_2 \qquad \boldsymbol{S}_1 \stackrel{def}{=} (a, r_r).\boldsymbol{S}_1 + (b, r_q).\boldsymbol{S}_2$$
$$\boldsymbol{P}_2 \stackrel{def}{=} (b, r_q).\boldsymbol{P}_2 + (c, r_s).\boldsymbol{P}_1 \qquad \boldsymbol{S}_2 \stackrel{def}{=} (d, r_z).\boldsymbol{S}_1$$
$$\boldsymbol{Q} \stackrel{def}{=} (c, r_s).\boldsymbol{Q} \qquad \boldsymbol{R} \stackrel{def}{=} (d, r_z).\boldsymbol{R}$$

$$\boldsymbol{Osc}(n) \stackrel{def}{=} (\mathbf{P}\{\boldsymbol{P}_1[20n]\} \underset{\{c\}}{\bowtie} \mathbf{Q}\{\boldsymbol{Q}[3n]\}) \underset{\{a,\,b\}}{\bowtie} (\mathbf{S}\{\boldsymbol{S}_1[20n]\} \underset{\{d\}}{\bowtie} \mathbf{R}\{\boldsymbol{R}[3n]\})$$

Figure 6.4: iGPEPA description of the model $\boldsymbol{Osc}(n)$.

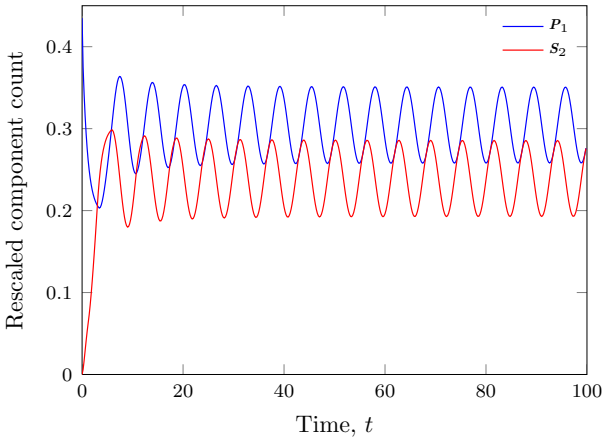


Figure 6.5: Solutions to the rescaled ODEs for the iGPEPA model $\boldsymbol{Osc}(n)$ given in Figure 6.4. Rates: $r_r = 1.0$, $r_q = 1.0$, $r_s = 3.0$, $r_z = 2.0$.

where the original rescaled ODE solutions can be recovered as follows: $\bar{v}_{\boldsymbol{P}_1}(t) = 10/23 - \hat{v}_1(t)$, $\bar{v}_{\boldsymbol{P}_2}(t) = \hat{v}_1(t)$, $\bar{v}_{\boldsymbol{S}_1}(t) = 10/23 - \hat{v}_2(t)$, $\bar{v}_{\boldsymbol{S}_2}(t) = \hat{v}_2(t)$, $\bar{v}_{\boldsymbol{Q}}(t) = 3/46$ and $\bar{v}_{\boldsymbol{R}}(t) = 3/46$. Figure 6.5 shows the quantities $v_{\boldsymbol{P}_1}(t)$ and $v_{\boldsymbol{S}_2}(t)$; the presence of a periodic limit cycle seems clear. We will discuss the implications of this for the steady-state convergence of the rescaled CTMCs in the next section.

6.3.3 Steady-state convergence theorem

Before presenting the convergence theorem we require a few dynamical systems definitions which we give here for the general time-autonomous differential equation $\dot{\mathbf{y}}(t) = \mathbf{k}(\mathbf{y}(t))$ where $\mathbf{y}(t) \in \mathbb{R}^n$ for $t \in \mathbb{R}_+$ and $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^n$. For more details see any standard text on dynamical systems, such as the book by Brin and Stuck [55] or by Wiggins [256].

Let $K \subseteq \mathbb{R}^n$ be compact, then the *omega limit set* of $\mathbf{w} \in K$ with respect to $\mathbf{y}(t)$, denoted $\omega(\mathbf{w}, \mathbf{y}(t))$, is the set of $\mathbf{z} \in \mathbb{R}^n$, such that $\lim_{k\to\infty} \mathbf{y}_{\mathbf{w}}(t_k) = \mathbf{z}$ for some sequence $\{t_k\}_{k=1}^{\infty}$ with $\lim_{k\to\infty} t_k = \infty$. The *Birkhoff centre* of $\mathbf{y}(t)$ in K, denoted $\mathcal{B}(\mathbf{y}(t), K)$, is the closure of the set of *recurrent points* in K, that is, the closure of the set of $\mathbf{w} \in K$ with $\mathbf{w} \in \omega(\mathbf{w}, \mathbf{y}(t))$. It is clear that this set contains any equilibrium points and periodic orbits reachable from K and in the case of an asymptotically stable (within K) fixed point $\mathbf{y}^* \in K$, $\mathcal{B}(\mathbf{y}(t), K) = \{\mathbf{y}^*\}$.

Now since K is compact it is a separable metric space under the restriction of the usual topology

on $\mathbb{R}^n$. Assume that $\mathbf{w} \in K$ implies that $\mathbf{y}_{\mathbf{w}}(t) \in K$ holds for all $t \in \mathbb{R}_+$. An *invariant measure* for $\mathbf{y}(t)$ is a probability measure μ on the measurable space $(K, \mathfrak{B}(K))^\dagger$ which is preserved by the differential equation at any time $t \in \mathbb{R}_+$. Formally, μ is an invariant measure if $\mu(\mathbf{y}_A^{-1}(t)) = \mu(A)$ for every $t \in \mathbb{R}_+$ and $A \in \mathfrak{B}(K)$, where $\mathbf{y}_A^{-1}(t) := \{\mathbf{w} \in K \ : \ \mathbf{y}_{\mathbf{w}}(t) \in A\}$. Fix $t > 0$. For such an invariant measure μ, the *Poincaré recurrence theorem* then says that for any $A \in \mathfrak{B}(K)$:

$$\mu(\{\mathbf{w} \in A \ : \ \exists N \geq 1 \text{ such that } \mathbf{y}_{\mathbf{w}}(nt) \notin A \text{ for all } n \geq N\}) = 0$$

Informally, this asserts that the set of points to which the differential equation eventually never returns at any time nt for $n \in \mathbb{Z}_+$ has measure zero.

Now since K is separable, there exists a countable basis $\{U_i\}_{i=1}^\infty$ for its topology. If for each i with $\mathbf{w} \in U_i$ and for any $N \geq 1$, there exists $n \geq N$ such that $\mathbf{y}_{\mathbf{w}}(nt) \in U_i$ then $\mathbf{w} \in \omega(\mathbf{w}, \mathbf{y}(t))$. Define then:

$$V_i := \{\mathbf{w} \in U_i \ : \ \forall N \geq 1, \exists n \geq N \text{ such that } \mathbf{y}_{\mathbf{w}}(nt) \in U_i\}$$

and observe that:

$$U_i = V_i \cup \{\mathbf{w} \in U_i \ : \ \exists N \geq 1 \text{ such that } \mathbf{y}_{\mathbf{w}}(nt) \notin U_i \text{ for all } n \geq N\}$$

so that we then have $\mu(V_i) = \mu(U_i)$ by the Poincaré recurrence theorem. Let also $K_i := V_i \cup (K \setminus U_i)$ and then $\mu(K_i) = \mu(K) = 1$. Finally $\cap_{i=1}^\infty K_i \subseteq \mathcal{B}(\mathbf{y}(t), K)$ so $\mu(\mathcal{B}(\mathbf{y}(t), K)) = 1$.

The steady-state analogue of Corollary 6.2.3 now follows. Its proof is based on techniques drawn from the field of discrete stochastic approximation algorithms, in particular the work of Benaïm [29] and can also be viewed as an extension of the mean-field stationary distribution convergence results of Benaïm and Le Boudec [30] from the discrete-time to the continuous-time case. The idea of the proof is to show that the limit of a sequence of stationary measures for the iGPEPA models is invariant with respect to the approximating ODE on a suitable compact set containing the ODE solutions and CTMC state spaces. We will then apply Poincaré's recurrence theorem as above.

Theorem 6.3.2. *Let $\{G^{(i)}\}_{i=1}^\infty$ be a sequence of structurally-equivalent* i*GPEPA models where $\mathcal{S}(G^{(i)}) \to \infty$ as $i \to \infty$. Again, assume the notation introduced in Section 6.1.1.*

Define the compact set:

$$K := \mathcal{R} \cap \left\{ \mathbf{x} \ : \ \mathbf{x} \in \mathbb{R}^{\mathcal{N}},\ x_i \geq 0,\ \sum_i x_i \leq 1 \right\}$$

where $\mathcal{R}$ is as defined by Equation 6.7.

Further assume that μ_i is a stationary measure of the rescaled CTMC of $G^{(i)}$ which we consider as a probability measure on $(K, \mathfrak{B}(K))$.[8]

Then if $C \subset K$ is closed and disjoint from $\mathcal{B}(\bar{\mathbf{v}}(t), K)$, we have that $\lim_{i \to \infty} \mu_i(C) = 0$.

[†] Where $\mathfrak{B}(K)$ is the Borel σ-algebra of K.

[8] This is possible because $\mathcal{R}$ contains the rescaled state space of all of the $G^{(i)}$ and the sum of their rescaled component counts are always bounded by 0 and 1.

Proof. First we note that the sequence of measures μ_i is tight[9] since the space K is compact. Therefore the sequence μ_i is also relatively compact [32, Theorem 6.1], meaning that every subsequence of μ_i, say μ_{ij} contains a further subsequence, say μ_{ij_k} which converges weakly. These limit points are not necessarily equal, that is, the sequence μ_i does not necessarily converge weakly itself. However, we will show that any limit of a subsequence of μ_i is at least an invariant measure for $\bar{\mathbf{v}}(t)$.

Let μ be some weak limit point, say $\mu_{ij} \xrightarrow{w} \mu$. So we wish to show that if $A \in \mathfrak{B}(K)$ and $t \in \mathbb{R}_+$, we have $\mu(\bar{\mathbf{v}}_A^{-1}(t)) = \mu(A)$. To see that μ is an invariant measure, it is sufficient [32, Theorem 1.3] to verify that for any bounded and continuous $g : K \to \mathbb{R}$:

$$\int_K g(\bar{\mathbf{v}}_{\mathbf{x}}(t))\,\mu(\mathrm{d}\mathbf{x}) = \int_K g(\mathbf{x})\,\mu(\mathrm{d}\mathbf{x}) \tag{6.12}$$

since $\int_K g(\mathbf{x})\,\mu(\bar{\mathbf{v}}_{\mathrm{d}\mathbf{x}}^{-1}(t)) = \int_K g(\bar{\mathbf{v}}_{\mathbf{x}}(t))\,\mu(\mathrm{d}\mathbf{x})$ [157, Lemma 1.22]. Before we proceed, we observe that for any i:

$$\int_K g(\mathbf{x})\,\mu_i(\mathrm{d}\mathbf{x}) = \int_K \mathbb{E}[g(\bar{\mathbf{N}}_{\mathbf{x}}^{(i)}(t))]\,\mu_i(\mathrm{d}\mathbf{x}) \tag{6.13}$$

This is true since the μ_i are stationary measures of the $\bar{\mathbf{N}}^{(i)}(t)$. To show Equation 6.12, fix $\delta > 0$, then combining Equation 6.13 with the fact that $\mu_{ij} \xrightarrow{w} \mu$, we may choose J sufficiently large such that for all $j \geq J$:

$$\left| \int_K g(\mathbf{x})\,\mu(\mathrm{d}\mathbf{x}) - \int_K \mathbb{E}[g(\bar{\mathbf{N}}_{\mathbf{x}}^{(i_j)}(t))]\,\mu_{ij}(\mathrm{d}\mathbf{x}) \right| < \delta/4 \tag{6.14}$$

We now proceed to bound the term $|\mathbb{E}[g(\bar{\mathbf{N}}_{\mathbf{x}}^{(i)}(t))] - g(\bar{\mathbf{v}}_{\mathbf{x}}(t))|$ for large enough i, uniformly for any rescaled state $\mathbf{x}$ of $G^{(i)}$. We note that g is uniformly continuous since K is compact. Therefore we can find $\alpha > 0$ independent of $\mathbf{x}$ such that for all $\mathbf{y} \in K$, $\|\mathbf{y} - \bar{\mathbf{v}}_{\mathbf{x}}(t)\| < \alpha \Rightarrow |g(\mathbf{y}) - g(\bar{\mathbf{v}}_{\mathbf{x}}(t))| < \delta/4$. This allows us to employ the following upper bound, where $\|g\|$ is an upper bound on the magnitude of g:

$$\left| \mathbb{E}[g(\bar{\mathbf{N}}_{\mathbf{x}}^{(i)}(t))] - g(\bar{\mathbf{v}}_{\mathbf{x}}(t)) \right| \leq \delta/4 + 2\|g\|\mathbb{P}\left\{ \sup_{0 \leq s \leq t} \|\bar{\mathbf{N}}_{\mathbf{x}}^{(i)}(s) - \bar{\mathbf{v}}_{\mathbf{x}}(s)\| \geq \alpha \right\}$$

which follows since $\delta/4$ is the maximum value of $|g(\bar{\mathbf{N}}_{\mathbf{x}}^{(i)}(t)) - g(\bar{\mathbf{v}}_{\mathbf{x}}(t))|$ on the event:

$$\left\{ \sup_{0 \leq s \leq t} \|\bar{\mathbf{N}}_{\mathbf{x}}^{(i)}(s) - \bar{\mathbf{v}}_{\mathbf{x}}(s)\| < \alpha \right\}$$

and its maximum value otherwise is bounded above by $2\|g\|$. Then applying Corollary 6.2.3, we may choose M sufficiently large such that for all $m \geq M$, $|\mathbb{E}[g(\bar{\mathbf{N}}_{\mathbf{x}}^{(m)}(t))] - g(\bar{\mathbf{v}}_{\mathbf{x}}(t))| \leq \delta/2$ uniformly for any $\mathbf{x}$ in the rescaled state space of $G^{(m)}$. Choose N sufficiently large such that $N \geq J$ and $i_N \geq M$, then, using this bound and that of Equation 6.14, we obtain for all $n \geq N$:

$$\begin{aligned} \left| \int_K g(\mathbf{x})\,\mu(\mathrm{d}\mathbf{x}) - \int_K g(\bar{\mathbf{v}}_{\mathbf{x}}(t))\,\mu_{i_n}(\mathrm{d}\mathbf{x}) \right| &\leq \left| \int_K g(\mathbf{x})\,\mu(\mathrm{d}\mathbf{x}) - \int_K \mathbb{E}[g(\bar{\mathbf{N}}_{\mathbf{x}}^{(i_n)}(t))]\,\mu_{i_n}(\mathrm{d}\mathbf{x}) \right| \\ &\quad + \int_K |\mathbb{E}[g(\bar{\mathbf{N}}_{\mathbf{x}}^{(i_n)}(t))] - g(\bar{\mathbf{v}}_{\mathbf{x}}(t))|\,\mu_{i_n}(\mathrm{d}\mathbf{x}) \\ &\leq 3\delta/4 \end{aligned} \tag{6.15}$$

[9] See for example the book by Billingsley [32, Page 37] for a definition of tightness of measures and also for the general theory of weak convergence of measures.

Also since $g(\bar{\mathbf{v}}_{\mathbf{x}}(t))$ is bounded and continuous as a function of $\mathbf{x}$,[†] we can use the fact that $\mu_{ij} \xrightarrow{w} \mu$ to find R sufficiently large such that for all $r \geq R$:

$$\left| \int_K g(\bar{\mathbf{v}}_{\mathbf{x}}(t))\, \mu(\mathrm{d}\mathbf{x}) - \int_K g(\bar{\mathbf{v}}_{\mathbf{x}}(t))\, \mu_{ir}(\mathrm{d}\mathbf{x}) \right| < \delta/4$$

Then combining this with Equation 6.15, we can obtain:

$$\left| \int_K g(\mathbf{x})\, \mu(\mathrm{d}\mathbf{x}) - \int_K g(\bar{\mathbf{v}}_{\mathbf{x}}(t))\, \mu(\mathrm{d}\mathbf{x}) \right| < \delta$$

Since δ was arbitrary, this verifies Equation 6.12 and thus shows that μ is an invariant measure for $\bar{\mathbf{v}}(t)$.

Now let $C \subset K$ be closed and disjoint from $\mathcal{B}(\bar{\mathbf{v}}(t), K)$. Then by the *Poincaré recurrence theorem*, $\mu(C) = 0$, which holds for all limit points μ of μ_i.

Finally, we assume for a contradiction that $\lim_{i\to\infty} \mu_i(C) \neq 0$. Then there is some $\epsilon > 0$ such that $\mu_{il}(C) > \epsilon$ for all elements of some subsequence μ_{il}. By relative compactness, we can find a further subsequence, say μ_{il_q}, such that $\mu_{il_q} \xrightarrow{w} \mu$ for some weak limit point μ. But since C is closed, we have $\limsup_{q\to\infty} \mu_{il_q}(C) \leq \mu(C) = 0$, a contradiction, which gives the required result. □

In the case of an asymptotically stable fixed point this theorem reduces to the following simpler corollary.

Corollary 6.3.3. *Assuming the setup of Theorem 6.3.2 and further that* $\mathcal{B}(\bar{\mathbf{v}}(t), K) = \{\bar{\mathbf{v}}^*\}$, *the sequence of measures* μ_n *converges in probability and thus also weakly in distribution to the point mass at* $\bar{\mathbf{v}}^*$.

Proof. For any $\epsilon > 0$, let U_ϵ be the open ϵ-ball around $\bar{\mathbf{v}}^*$. Then, clearly by Theorem 6.3.2, $\mu_n(K \backslash U_\epsilon) \to 0$ as $n \to \infty$. The required result then follows immediately since ϵ is arbitrary. □

In the case of the client–server model, recall that the matrix $\mathbf{P}$ given by Equation 6.11 is a solution to the system of LMIs of Equation 6.10 in the case of rate parameters: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.2$, $r_b = 0.1$, $r_d = 1.0$, $r_{rst} = 1.0$. Therefore we may use Theorem 6.3.1 to verify asymptotic stability of the unique fixed point $(0.1685, 0.0830, 0.4151, 0.0539, 0.2491, 0.0303)^T$. Corollary 6.3.3 can then be applied to show that the stationary measures of the rescaled CTMCs underlying the sequence of models $\{\boldsymbol{CS}(2n, n)\}_{n=1}^{\infty}$ are guaranteed to converge to the point mass at this fixed point. This is illustrated by Figure 6.6.

In the case of the model $\boldsymbol{Osc}(n)$ given by Figure 6.4, the situation is more complicated. Figure 6.5 and additional numerical integration of the associated ODEs from many different initial conditions suggest that the Birkhoff centre of the associated ODEs consists just of the periodic limit cycle depicted in Figure 6.5. However we emphasise that any number of numerical integrations does not alone constitute formal proof of this. If we are to accept that this is the case then Theorem 6.3.2 can be used to establish that in the limit, the sequence of stationary measures should be concentrated on this limit cycle. This is indeed supported by simulation results shown in Figure 6.7.

[†]See for example the book by Miller and Michel [187] for the relevant arguments regarding continuous dependence on initial conditions.

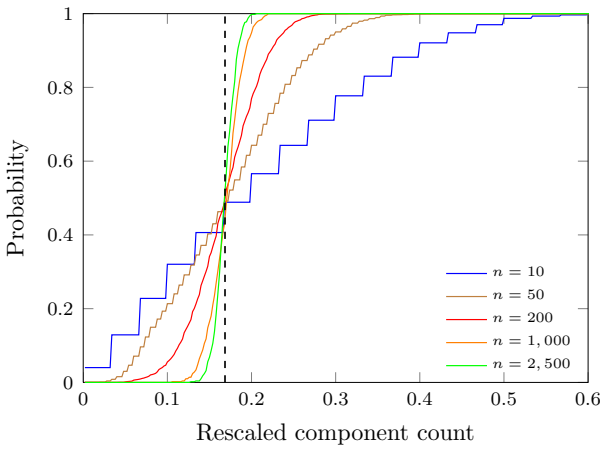


Figure 6.6: Comparison of ***Client*** component count steady-state CDF (computed by simulation) with ODE fixed-point approximation for the model $\boldsymbol{CS}(2n, n)$. Rates: $r_r = 2.0$, $r_{tmt} = 0.3$, $r_t = 0.2$, $r_b = 0.1$, $r_d = 1.0$, $r_{rst} = 1.0$.

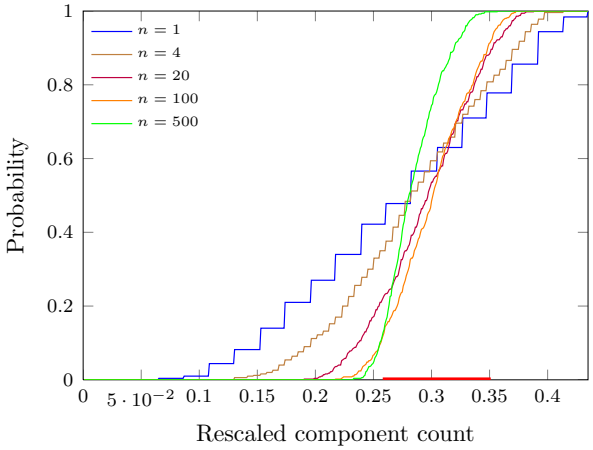


Figure 6.7: Comparison of $\boldsymbol{P}_1$ component count steady-state CDF (computed by simulation) with the ODE Birkhoff centre (the solid red line on the x-axis) for the model $\boldsymbol{Osc}(n)$. Rates: $r_r = 1.0$, $r_q = 1.0$, $r_s = 3.0$, $r_z = 2.0$.

It is our experience that most systems of ODEs obtained from *i*GPEPA models constructed to model real systems have asymptotically stable fixed points — a fact we can usually verify formally with the techniques discussed in Section 6.3.2. For this reason, in this thesis, we have not explored techniques for proving formally the existence of periodic limit cycles or that these constitute the Birkhoff centre exactly in cases such as that of $\boldsymbol{Osc}(n)$.

By way of other related work, we note finally that Ding [91] has also investigated the steady-state convergence problem for PEPA models. Specifically Ding [91] gives a condition on the underlying CTMCs of a sequence of PEPA models with increasing total component population. If it can be verified, this condition is shown to guarantee simultaneously both asymptotic stability of the ODE fixed point and convergence of the rescaled component-count expectations to the ODE solution. Unfortunately this condition is specified directly in terms of the steady-state probabilities of the underlying CTMCs so, as recognised by the author, its verification is not scalable due to the state-space explosion problem. Therefore it appears just as computationally (in)feasible as to compute exactly the steady-state probability vector for the specific model of interest. As it stands then, Ding's condition does not offer a *useful* guarantee of the validity of the ODE approximation in the steady-state limit.

Supported by empirical numerical experiments on a case study, Ding [91, Conjecture 1] also conjectures that the condition always holds for any PEPA model and thus that verifying it is unnecessary in practice. Unfortunately, this is untrue — our model $\boldsymbol{Osc}(n)$ provides a counterexample as noted at the end of Section 6.3.2. In addition to the general condition, Ding [91] considers a particular class of simple model. In the GPEPA framework, this would be stated as the class of models with two component groups and cooperation on one timed action type between them. For these models, Ding [91] is able to show asymptotic stability of the associated ODEs' fixed point (under certain conditions on the initial population sizes). However it is not shown, even for this class of models, that this implies that the limit of the stationary measures is concentrated on the fixed point.

6.4 Convergence of passage-time approximations

In this section, we present convergence results for global and individual passage times to their differential-equation approximations which were given in Chapter 4. These results build directly on the generic convergence results of the last two sections (Corollaries 6.2.3 and 6.3.3).

6.4.1 Global passage times

Let $\{G^{(i)}\}_{i=1}^{\infty}$ be a sequence of structurally-equivalent *i*GPEPA models such that $\mathcal{S}(G^{(i)}) \to \infty$ as $i \to \infty$. We will assume that the global passage-time measure that we are considering is specified by probes as in Section 5.6. In particular, we consider a sequence of global passage-time measurements as given by the sequence of measure specifications on each model $G^{(i)}$:

$$\begin{array}{l} \epsilon : \mathsf{start}, R_g^{(i)} : \mathsf{stop} \\ \mathsf{observes} \quad \{Probe_l\} \\ \mathsf{where} \quad \{Location\} \quad \mathsf{in} \quad G^{(i)} \end{array} \tag{6.16}$$

Recall that the global probes must belong to the restricted grammar of Equation 5.23 and that any local probes are applied to each component in a group so that, after applying the local probes, we are left with a new sequence of structurally-equivalent iGPEPA models, say $\{\tilde{G}^{(i)}\}_{i=1}^{\infty}$, also with the property that $\mathcal{S}(\tilde{G}^{(i)}) \to \infty$ as $i \to \infty$. Write $\dot{\mathbf{v}}(t) = \mathbf{f}(\mathbf{v}(t))$ for the associated system of component-counting ODEs, and $\dot{\mathbf{w}}(t) = \mathbf{g}(\mathbf{v}(t))$ for the associated system of action-counting ODEs.

In order for the passage-time measurements to be comparable, that is, to have the same ODE approximation, we require that the $R_g^{(i)}$ have identical structure except that the atoms in $R_g^{(i)}$ are each of the form $(a_1 \mid \ldots \mid a_k)[n \times \mathcal{S}(\tilde{G}^{(i)})]$ where the only dependence on i is through $\mathcal{S}(\tilde{G}^{(i)})$. Similarly, we also require that any integers occuring in state guard predicate expressions have the form $n \times \mathcal{S}(\tilde{G}^{(i)})$.

Theorem 6.4.1. *Let $\{G^{(i)}\}_{i=1}^{\infty}$ be a sequence of structurally-equivalent* i*GPEPA models where $\mathcal{S}(G^{(i)}) \to \infty$ as $i \to \infty$. Assume further the setup and notation introduced above.*

Then the ODE point-mass approximation $\mathcal{U}(\tilde{G}^{(i)}, \mathsf{V}_t^{(i)}, \mathsf{W}_t^{(i)}, R_g^{(i)}, 0)$ to the global passage-time random variable $\mathcal{U}(\tilde{G}^{(i)}, \mathsf{N}_t^{(i)}, \mathsf{A}_t^{(i)}, R_g^{(i)}, 0)$ is independent of i, so write it as, say γ. Further if $\gamma < \infty$, as $i \to \infty$, we have for any $\epsilon > 0$, $\mathbb{P}\{|\mathcal{U}(\tilde{G}^{(i)}, \mathsf{N}_t^{(i)}, \mathsf{A}_t^{(i)}, R_g^{(i)}, 0) - \gamma| > \epsilon\} \to 0$.

Proof. It is sufficient to show that if $\{e^{(i)}\}_{i=1}^{\infty}$ is a sequence of random variables that converges in probability to v as $i \to \infty$, which we write $e^{(i)} \xrightarrow{P} v$, then $\mathcal{U}(\tilde{G}^{(i)}, \mathsf{N}_t^{(i)}, \mathsf{A}_t^{(i)}, R_g^{(i)}, e^{(i)}) \xrightarrow{P} \gamma$ where $\gamma := \mathcal{U}(\tilde{G}^{(i)}, \mathsf{V}_t^{(i)}, \mathsf{W}_t^{(i)}, R_g^{(i)}, v)$ and is independent of i. We will proceed by induction on the structure of $R_g^{(i)}$.

Case $R_g^{(i)} = (a_1 \mid \ldots \mid a_k)[n \times \mathcal{S}(\tilde{G}^{(i)})]$

In this case:

$$\begin{aligned}
\mathcal{U}(\tilde{G}^{(i)}, \mathsf{V}_t^{(i)}, \mathsf{W}_t^{(i)}, R_g^{(i)}, v) &= \inf\left\{ t \geq v \;:\; \sum_{j=1}^{k} w_{a_j}^{(i)}(t) - \sum_{j=1}^{k} w_{a_j}^{(i)}(v) \geq n \times \mathcal{S}(\tilde{G}^{(i)}) \right\} \\
&= \inf\left\{ t \geq v \;:\; \sum_{j=1}^{k} \bar{w}_{a_j}(t) - \sum_{j=1}^{k} \bar{w}_{a_j}(v) \geq n \right\} \\
&=: \gamma
\end{aligned}$$

which is thus independent of i. Now assume that:

$$\mathcal{U}(\tilde{G}^{(i)}, \mathsf{N}_t^{(i)}, \mathsf{A}_t^{(i)}, R_g^{(i)}, e^{(i)}) = \inf\left\{ t \geq e^{(i)} \;:\; \sum_{j=1}^{k} \bar{A}_{a_j}^{(i)}(t) - \sum_{j=1}^{k} \bar{A}_{a_j}^{(i)}(e^{(i)}) \geq n \right\} < \gamma - \epsilon$$

Then we have some $e^{(i)} \leq t < \gamma$ for which:

$$\sum_{j=1}^{k} \bar{A}_{a_j}^{(i)}(t) - \sum_{j=1}^{k} \bar{A}_{a_j}^{(i)}(e^{(i)}) \geq n$$

But $\sum_{j=1}^{k} \bar{w}_{a_j}(t) - \sum_{j=1}^{k} \bar{w}_{a_j}(v) = n - \delta$ for some $\delta > 0$, so:

$$\left| \sum_{j=1}^{k} \bar{A}_{a_j}^{(i)}(t) - \sum_{j=1}^{k} \bar{w}_{a_j}(t) \right| + \left| \sum_{j=1}^{k} \bar{A}_{a_j}^{(i)}(e^{(i)}) - \sum_{j=1}^{k} \bar{w}_{a_j}(v) \right| \geq \delta$$

It then follows by Corollary 6.2.3 that this occurs with vanishing probability as $i \to \infty$. The case $\mathcal{U}(\tilde{G}^{(i)}, \mathsf{N}_t^{(i)}, \mathsf{A}_t^{(i)}, R_g^{(i)}, e^{(i)}) > \gamma - \epsilon$ is proved similarly, which yields the required result.

Case $R_g^{(i)} = R_g^1; R_g^2$

In this case, by induction, we have $\mathcal{U}(\tilde{G}^{(i)}, \mathsf{N}_t^{(i)}, \mathsf{A}_t^{(i)}, R_g^1, e^{(i)}) \xrightarrow{P} \mathcal{U}(\tilde{G}^{(i)}, \mathsf{V}_t^{(i)}, \mathsf{W}_t^{(i)}, R_g^1, v)$ and $\mathcal{U}(\tilde{G}^{(i)}, \mathsf{N}_t^{(i)}, \mathsf{A}_t^{(i)}, R_g^2, e^{(i)}) \xrightarrow{P} \mathcal{U}(\tilde{G}^{(i)}, \mathsf{V}_t^{(i)}, \mathsf{W}_t^{(i)}, R_g^2, v)$, both of which limits are independent of i. The result then follows immediately by continuity of the maximum function.

Case $R_g^{(i)} = R_g^1 \mid R_g^2$

This proof of this case is essentially identical to that of the previous.

Case $R_g^{(i)} = R_g^1, R_g^2$

In this case, by induction, we have that $\mathcal{U}(\tilde{G}^{(i)}, \mathsf{N}_t^{(i)}, \mathsf{A}_t^{(i)}, R_g^1, e^{(i)}) \xrightarrow{P} \mathcal{U}(\tilde{G}^{(i)}, \mathsf{V}_t^{(i)}, \mathsf{W}_t^{(i)}, R_g^1, v)$ with the limit independent of i. It thus also follows by induction that:

$$\mathcal{U}(\tilde{G}^{(i)}, \mathsf{N}_t^{(i)}, \mathsf{A}_t^{(i)}, R_g^2, \mathcal{U}(\tilde{G}^{(i)}, \mathsf{N}_t^{(i)}, \mathsf{A}_t^{(i)}, R_g^1, e^{(i)})) \xrightarrow{P} \mathcal{U}(\tilde{G}^{(i)}, \mathsf{V}_t^{(i)}, \mathsf{W}_t^{(i)}, R_g^2, \mathcal{U}(\tilde{G}^{(i)}, \mathsf{V}_t^{(i)}, \mathsf{W}_t^{(i)}, R_g^1, v))$$

with the limit again independent of i.

Case $R_g^{(i)} = \{pred^{(i)}\} R_g$

Let $pred_*^{(i)}(t)$ be the state predicate in terms of the ODE approximations. Note that by dividing through each relational expression in the predicate by $\mathcal{S}(\tilde{G}^{(i)})$, we can see that $pred_*^{(i)}(t)$ and thus the quantity $\inf\{t \geq v \ : \ pred_*^{(i)}(t)\}$ is independent of i. Furthermore, we can prove similarly to in the first case that $\inf\{t \geq e^{(i)} \ : \ pred^{(i)}(t)\} \xrightarrow{P} \inf\{t \geq v \ : \ pred_*^{(i)}(t)\}$. The result then follows by induction. □

An example of the convergence stated in this theorem can be seen for the client–server model in Figure 4.2 and for the distributed wireless network model in Figure 5.14.

6.4.2 Individual passage times

We turn now to proving convergence results for steady-state individual passage-time measures.

Let $\{G^{(i)}\}_{i=1}^{\infty}$ be a sequence of structurally-equivalent iGPEPA models such that $\mathcal{S}(G^{(i)}) \to \infty$ as $i \to \infty$ and with associated ODE $\dot{\mathbf{v}}(t) = \mathbf{f}(\mathbf{v}(t))$. We will assume that the steady-state individual passage-time measure that we are considering is specified by probes as in Section 5.5.1. In particular, we consider a sequence of individual passage-time measurements as given by the sequence of measure specifications on each model $G^{(i)}$:

$$\begin{aligned} &\mathsf{begin} : \mathsf{start}, \mathsf{end} : \mathsf{stop}^{\hookrightarrow} \\ &\mathsf{observes} \quad Pb \stackrel{def}{=} R_l : \mathsf{begin}, R_l : \mathsf{end}^{\hookrightarrow} \\ &\mathsf{where} \quad H\{P[?n]\} \Longrightarrow H\{(P \underset{*}{\bowtie} Pb) \wr\wr P[?n-1]\} \\ &\mathsf{in} \quad G^{(i)} \end{aligned} \tag{6.17}$$

Let $\tilde{G}^{(i)}$ be the sequence of iGPEPA models obtained by applying the location specification in Equation 6.17 to each model $G^{(i)}$. As in Section 5.5.1, in order for each passage-time measure to be meaningful, we require here that the underlying CTMC of each $G^{(i)}$ has a unique stationary distribution. Let also $\tilde{G}'^{(i)}$ be the sequence of iGPEPA models obtained by applying the location specification in Equation 6.17 to each model $G^{(i)}$ this time for the absorbing version of the local probe $Pb' \stackrel{def}{=} R_l : \mathsf{begin}, R_l : \mathsf{end}$.

Recall that in line with Section 5.5.1, we consider each model $\tilde{G}^{(i)}$ evolving in its stationary regime and then at arbitrary fixed time 0, its evolution is switched to that of $\tilde{G}'^{(i)}$, initialised by the current state of $\tilde{G}^{(i)}$. Then by timing how long it takes for the absorbing probe Pb' in $\tilde{G}'^{(i)}$ to enter its absorbed state, say, Pb_{Acc}, we can measure the duration of the passage time of interest. Define $C^{(i)}(t) \in ds^*(P \underset{*}{\bowtie} Pb')$ to be the stochastic process, which tracks the state of the probed fluid component in the group H as the model $\tilde{G}'^{(i)}$ evolves. Then the sequence of passage-time CDFs of interest is given by:

$$\mathbb{P}\{C^{(i)}(t) \in - \underset{*}{\bowtie} Pb_{\mathrm{Acc}} \mid C^{(i)}(0) \in \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb), C^{(i)}(0-) \notin \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb)\}$$

Write $\tilde{\mathcal{B}} := \mathcal{B}(\tilde{G}^{(i)})$ and $\tilde{\mathcal{B}}' := \mathcal{B}(\tilde{G}'^{(i)})$, both of which are independent of i. Before proving the convergence theorem, we first develop the ODE approximation in the context of the sequences of models and show that, similarly to previous results, the approximation is the same for all models in the sequence. We will find the following observation useful. For $\tilde{\mathsf{V}}^{(i)} \in \tilde{\mathcal{B}} \to \mathbb{R}_+$, the component rate function (Definition 3.2.4) has the following form for fluid component derivative states $R \in ds^*(P \underset{*}{\bowtie} Pb)$:

$$\mathcal{R}_\alpha(\tilde{G}^{(i)}, \tilde{\mathsf{V}}^{(i)}, H, R) = \mathcal{R}^{(i)}_{H,R} \times \tilde{\mathsf{V}}^{(i)}(H, R) \tag{6.18}$$

where $\mathcal{R}^{(i)}_{H,R}$ depends on $\tilde{\mathsf{V}}^{(i)}$ only through the rescaled component counts:

$$\begin{aligned} \frac{1}{\mathcal{S}(G^{(i)})} \left(\tilde{\mathsf{V}}^{(i)}(H,U) + \sum_{O \in U \underset{*}{\bowtie} -} \tilde{\mathsf{V}}^{(i)}(H,O) \right) \quad &\text{for } (H,U) \in \mathcal{B} \\ \frac{1}{\mathcal{S}(G^{(i)})} \tilde{\mathsf{V}}^{(i)}(Y,U) \quad &\text{for } (Y,U) \in \mathcal{B}, Y \neq H \end{aligned} \tag{6.19}$$

In order for the ODE approximation to these passage times to be well defined, we assume that the system of ODEs $\dot{\mathbf{v}}(t) = \mathbf{f}(\mathbf{v}(t))$ has a unique rescaled fixed point that is meaningful in the sense that it lies in K (as defined in Theorem 6.3.2). We represent this rescaled fixed point by $\bar{\mathsf{V}} \in \mathcal{B} \to \mathbb{R}_+$.

We will write $\dot{\tilde{\mathbf{v}}}(t) = \tilde{\mathbf{f}}(\tilde{\mathbf{v}}(t))$ for the ODEs associated to any $\tilde{G}^{(i)}$ (it is straightforward to see that the function $\tilde{\mathbf{f}}$ is the same for all i). A fixed point, say $\tilde{\mathsf{V}}^{(i)} \in \tilde{\mathcal{B}} \to \mathbb{R}_+$, for this system is meaningful in the context of the model $\tilde{G}^{(i)}$ if $\bar{\mathsf{V}}^{(i)} \in \mathcal{B} \to \mathbb{R}_+$ defined for $(Y, Q) \in \mathcal{B}$ by:

$$\bar{\mathsf{V}}^{(i)}(Y,Q) := \frac{1}{\mathcal{S}(G^{(i)})} \begin{cases} \tilde{\mathsf{V}}^{(i)}(H,Q) + \sum_{R \in Q \underset{*}{\bowtie} _} \tilde{\mathsf{V}}^{(i)}(H,R) & : Y = H \\ \tilde{\mathsf{V}}^{(i)}(Y,Q) & : \text{otherwise} \end{cases}$$

lies in K; and for each $R \in ds^*(P \underset{*}{\bowtie} Pb)$, $\tilde{\mathsf{V}}^{(i)}(H,R) \geq 0$ and $\sum_{R \in ds^*(P \underset{*}{\bowtie} Pb)} \tilde{\mathsf{V}}^{(i)}(H,R) = 1$. We show now how such a fixed point $\tilde{\mathsf{V}}^{(i)}$ can be constructed from $\bar{\mathsf{V}}$. Let $Q \in ds^*(P \underset{*}{\bowtie} Pb)$ and note that, in this case, by the above observation regarding the component rate function:

$$\tilde{f}_{H,Q}(\tilde{\mathsf{V}}^{(i)}) = \sum_{R \in ds^*(P \underset{*}{\bowtie} Pb)} A^{(i)}_{R,Q} \times \tilde{\mathsf{V}}^{(i)}(H,R) \tag{6.20}$$

where the $A^{(i)}_{R,Q}$ depend on $\tilde{\mathsf{V}}^{(i)}$ only through the rescaled component counts given in Equation 6.19. It thus makes sense to equate the two quantities of Equation 6.19 with $\bar{\mathsf{V}}(H,U)$ and $\bar{\mathsf{V}}(Y,U)$, respectively and we may thus write $A_{R,Q}$ independently of i. Then setting Equation 6.20 to zero for each $Q \in ds^*(P \underset{*}{\bowtie} Pb)$ forms a linear system of equations, say $\mathbf{x}^T\mathbf{A} = \mathbf{0}$, where $\mathbf{A} := (A_{R,Q})$ is the $|ds^*(P \underset{*}{\bowtie} Pb)| \times |ds^*(P \underset{*}{\bowtie} Pb)|$ real matrix of this linear system which is easily seen to have the form of a generator matrix of a CTMC. Indeed, it is the matrix of the 'marginal CTMC' underlying $P \underset{*}{\bowtie} Pb$ in the context of the model $\tilde{G}^{(i)}$ where the rates of actions on which the component cooperates with the rest of the model are given by the values of the relevant components of the fixed point $\bar{\mathsf{V}}$. For this reason, this linear system must have a unique solution that is a probability vector or, otherwise, the assumption that the CTMC underlying $\tilde{G}^{(i)}$ has a unique stationary distribution would be contradicted. For $Q \in ds^*(P \underset{*}{\bowtie} Pb)$, we thus define $\tilde{\mathsf{V}}^{(i)}(H,Q)$ to be this unique probability vector solution. Together with the above, this fully determines the quantity $\tilde{\mathsf{V}}^{(i)}$. It is straightforward to verify that this is indeed a meaningful fixed point of the system $\dot{\tilde{\mathbf{v}}}(t) = \tilde{\mathbf{f}}(\tilde{\mathbf{v}}(t))$.

For a given $\tilde{G}^{(i)}$, it is also clear that each distinct meaningful fixed point of $\dot{\tilde{\mathbf{v}}}(t) = \tilde{\mathbf{f}}(\tilde{\mathbf{v}}(t))$ yields a distinct meaningful fixed point of $\dot{\mathbf{v}}(t) = \mathbf{f}(\mathbf{v}(t))$, so, for each $\tilde{G}^{(i)}$, $\dot{\tilde{\mathbf{v}}}(t) = \tilde{\mathbf{f}}(\tilde{\mathbf{v}}(t))$ must have a unique meaningful fixed point. Furthermore, for $Q \in ds^*(P \underset{*}{\bowtie} Pb)$, we observe that the quantities $\tilde{\mathsf{V}}^{(i)}(H,Q)$ were in fact constructed independently of i and thus may be written as $\tilde{\mathsf{V}}(H,Q)$.

In order to proceed with the ODE approximation, we need to compute the approximation to the transition-instant distributions as given in Equation 5.17 of Section 5.5.1. Specifically, we see that for $Q \in ds^*(P \underset{*}{\bowtie} Pb)$, the approximation is given by:

$$\begin{aligned} &\mathbb{P}\{C^{(i)}(0) = Q \mid C^{(i)}(0) \in \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb), C^{(i)}(0-) \notin \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb)\} \approx \\ &\qquad \frac{\sum_{\alpha \in \mathcal{A}_t} \sum_{C \in ds^*(P \underset{*}{\bowtie} Pb)} \left(\frac{1}{r_\alpha(C)} \sum_{C \xrightarrow{(\alpha,\lambda),(\ldots,\mathsf{begin},\ldots)} Q} \lambda \times \mathcal{R}_\alpha(\tilde{G}^{(i)}, \tilde{\mathsf{V}}^{(i)}, H, C) \right)}{\sum_{\alpha \in \mathcal{A}_t} \sum_{C \in ds^*(P \underset{*}{\bowtie} Pb)} \left(\frac{1}{r_\alpha(C)} \sum_{C \xrightarrow{(\alpha,\lambda),(\ldots,\mathsf{begin},\ldots)} } \lambda \times \mathcal{R}_\alpha(\tilde{G}^{(i)}, \tilde{\mathsf{V}}^{(i)}, H, C) \right)} \end{aligned} \tag{6.21}$$

Now by the earlier observation, this expression also depends on $\tilde{\mathsf{V}}^{(i)}$ only through the rescaled component counts of Equation 6.19 and the quantities $\tilde{\mathsf{V}}(H,R)$ for $R \in ds^*(P \underset{*}{\bowtie} Pb)$. It is thus independent of i.

Finally, in order to compute the approximate CDF, we recall from Section 5.5.1, that we must solve the ODEs corresponding to $\tilde{G}'^{(i)}$, with initial conditions derived appropriately from $\tilde{\mathsf{V}}^{(i)}$. In this manner, let $\tilde{\mathsf{V}}'^{(i)}_t \in \tilde{\mathcal{B}}' \to \mathbb{R}_+$ represent the unique solution to these ODEs with the initial condition $\tilde{\mathsf{V}}'^{(i)}_0(H,Q)$ for $Q \in ds^*(P \underset{*}{\bowtie} Pb')$ given by the quantity of Equation 6.21 (this is zero if $Q \notin ds^*(P \underset{*}{\bowtie} Pb)$). For all other $(Y,R) \in \tilde{\mathcal{B}}'$, the initial value $\tilde{\mathsf{V}}'^{(i)}_0(Y,R)$ is set to $\tilde{\mathsf{V}}^{(i)}(Y,R)$. Note that in the limit as $i \to \infty$, we have for each $(Y,Q) \in \mathcal{B}$:

$$\bar{\mathsf{V}}(Y,Q) \longleftarrow \frac{1}{\mathcal{S}(G^{(i)})} \begin{cases} \tilde{\mathsf{V}}'^{(i)}_0(H,Q) + \sum_{R \in (Q \underset{*}{\bowtie} —)_—} \tilde{\mathsf{V}}'^{(i)}_0(H,R) & : Y = H \\ \tilde{\mathsf{V}}'^{(i)}_0(Y,Q) & : \text{otherwise} \end{cases}$$

and thus by Grönwall's inequality [e.g. 97, Page 498], we also have for $t \in [0,T]$, uniformly as $i \to \infty$:

$$\bar{\mathsf{V}}(Y,Q) \longleftarrow \frac{1}{\mathcal{S}(G^{(i)})} \begin{cases} \tilde{\mathsf{V}}'^{(i)}_t(H,Q) + \sum_{R \in (Q \underset{*}{\bowtie} —)_—} \tilde{\mathsf{V}}'^{(i)}_t(H,R) & : Y = H \\ \tilde{\mathsf{V}}'^{(i)}_t(Y,Q) & : \text{otherwise} \end{cases} \tag{6.22}$$

Finally, we observe that for $R \in ds^*(P \underset{*}{\bowtie} Pb')$, the corresponding ODE has the form:

$$\dot{\tilde{\mathsf{V}}}'^{(i)}_t(H,R) = \sum_{U \in ds^*(P \underset{*}{\bowtie} Pb')} C^{(i)}_{t,U,R} \times \tilde{\mathsf{V}}'^{(i)}_t(H,U) \tag{6.23}$$

where, in the limit $i \to \infty$, the $C^{(i)}_{t,U,R}$ depend on $\tilde{\mathsf{V}}'^{(i)}_t$ only through the constant rescaled quantities of Equation 6.22, and can thus be written as $C_{U,R}$ independent of both i and t. Therefore, again by Grönwall's inequality, the quantities $\tilde{\mathsf{V}}'^{(i)}_t(H,R)$ have a limit as $i \to \infty$ uniformly for all $t \in [0,T]$, which we write as $\tilde{\mathsf{V}}'_t(H,R)$. It then follows immediately that the CDF approximation $\sum_{C \in — \underset{*}{\bowtie} Pb_{\mathrm{Acc}}} \tilde{\mathsf{V}}'_t(H,C)$ is thus independent of i for all $t \in [0,T]$.

Theorem 6.4.2. *Let $\{G^{(i)}\}_{i=1}^{\infty}$ be a sequence of structurally-equivalent* i*GPEPA models where $\mathcal{S}(G^{(i)}) \to \infty$ as $i \to \infty$. Assume the setup and notation introduced above.*

Fix $T > 0$. Further assume that the system of ODEs $\dot{\mathbf{v}}(t) = \mathbf{f}(\mathbf{v}(t))$ has an asymptotically stable fixed point within K (as defined in Theorem 6.3.2), that is, $|\mathcal{B}(\mathbf{v}(t),K)| = 1$.

Then we have uniformly for any $t \in [0,T]$, as $i \to \infty$:

$$\mathbb{P}\{C^{(i)}(t) \in — \underset{*}{\bowtie} Pb_{Acc} \mid C^{(i)}(0) \in \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb), C^{(i)}(0-) \notin \mathcal{S}_{\mathsf{begin}}(P \underset{*}{\bowtie} Pb)\} \longrightarrow \sum_{C \in — \underset{*}{\bowtie} Pb_{Acc}} \tilde{\mathsf{V}}'_t(H,C)$$

Proof. First we note that the CTMC underlying each $G^{(i)}$ has a unique stationary distribution since this is true of each $\tilde{G}^{(i)}$. Let $\bar{\mathsf{N}}^{(i)} \in \mathcal{B} \to \mathbb{R}_+$ be the random variable given by the rescaled component counts of $G^{(i)}$ at time 0. Then by Theorem 6.3.2, as $i \to \infty$, we have that $\bar{\mathsf{N}}^{(i)}$ converges in probability to the ODE fixed point $\bar{\mathsf{V}}$. Let $\tilde{\mathsf{N}}^{(i)} \in \tilde{\mathcal{B}} \to \mathbb{R}_+$ be the random variable

given by the component counts of $\tilde{G}^{(i)}$ at time 0. Also define $\bar{\tilde{\mathsf{N}}}^{(i)} \in \mathcal{B} \to \mathbb{R}_+$ as follows, for $(Y,Q) \in \mathcal{B}$:

$$\bar{\tilde{\mathsf{N}}}^{(i)}(Y,Q) := \begin{cases} \frac{1}{\mathcal{S}(G^{(i)})}\left(\tilde{\mathsf{N}}^{(i)}(H,Q) + \sum_{R \in Q \underset{*}{\rtimes\!\!\ltimes} -} \tilde{\mathsf{N}}^{(i)}(H,R)\right) & : Y = H \\ \frac{1}{\mathcal{S}(G^{(i)})}\tilde{\mathsf{N}}^{(i)}(Y,Q) & : \text{otherwise} \end{cases}$$

Then since $\bar{\tilde{\mathsf{N}}}^{(i)}(Y,Q)$ is equal in distribution to $\bar{\mathsf{N}}^{(i)}(Y,Q)$, it is immediate that $\bar{\tilde{\mathsf{N}}}^{(i)}(Y,Q)$ converges in probability to $\bar{\mathsf{V}}(Y,Q)$.

We will now proceed to show that this convergence in probability still holds when conditioned on a begin-signal transition having just occurred at time 0. For $(Y,R) \in \mathcal{B}$, the expectation of $\bar{\tilde{\mathsf{N}}}^{(i)}(Y,R)$ conditioned on such a transition can be computed as [e.g. 218, Definition 96 and Proposition 97]:

$$\frac{\sum_{\alpha \in \mathcal{A}_t} \sum_{C \in ds^*(P \underset{*}{\rtimes\!\!\ltimes} Pb)} \left(\frac{1}{r_\alpha(C)} \sum_{C \xrightarrow{(\alpha,\lambda),(\ldots,\mathsf{begin},\ldots)}} \lambda \times \mathbb{E}[(\bar{\tilde{\mathsf{N}}}^{(i)}(Y,R) + S)\mathcal{R}_\alpha(\tilde{G}^{(i)}, \tilde{\mathsf{N}}^{(i)}, H, C)]\right)}{\sum_{\alpha \in \mathcal{A}_t} \sum_{C \in ds^*(P \underset{*}{\rtimes\!\!\ltimes} Pb)} \left(\frac{1}{r_\alpha(C)} \sum_{C \xrightarrow{(\alpha,\lambda),(\ldots,\mathsf{begin},\ldots)}} \lambda \times \mathbb{E}[\mathcal{R}_\alpha(\tilde{G}^{(i)}, \tilde{\mathsf{N}}^{(i)}, H, C)]\right)}$$

where $|S| \leq 1/\mathcal{S}(G^{(i)})$. By recalling the observation of Equation 6.18, it is straightforward to see that this converges to $\bar{\mathsf{V}}(Y,R)$ as $i \to \infty$. Similarly it is straightforward to see that the variance of $\bar{\tilde{\mathsf{N}}}^{(i)}(Y,R)$ conditioned on a begin-signal transition having just occurred converges to zero as $i \to \infty$. It thus follows that the desired convergence in probability still holds conditioned on such a transition.

Next we wish to show that for $Q \in ds^*(P \underset{*}{\rtimes\!\!\ltimes} Pb)$, $\mathbb{E}[\tilde{\mathsf{N}}^{(i)}(H,Q)]$ converges to $\tilde{\mathsf{V}}(H,Q)$ as $i \to \infty$. Dynkin's formula [e.g. 210, Page 254] gives:

$$\mathbb{E}[\tilde{f}_{H,Q}(\tilde{\mathsf{N}}^{(i)})] = 0$$

and we recall that due to the form of $\tilde{\mathbf{f}}$, we may write this as:

$$\sum_{R \in ds^*(P \underset{*}{\rtimes\!\!\ltimes} Pb)} \mathbb{E}[B^{(i)}_{R,Q} \times \tilde{\mathsf{N}}^{(i)}(H,R)] = 0$$

where the $B^{(i)}_{R,Q}$ depend on $\tilde{\mathsf{N}}^{(i)}$ only through the rescaled quantities $\bar{\tilde{\mathsf{N}}}^{(i)}$. It thus follows that:

$$\sum_{R \in ds^*(P \underset{*}{\rtimes\!\!\ltimes} Pb)} A_{R,Q} \times \mathbb{E}[\tilde{\mathsf{N}}^{(i)}(H,R)] \longrightarrow 0$$

as $i \to \infty$, where the matrix $\mathbf{A} := (A_{R,Q})$ is as defined earlier in Equation 6.20. Then since the linear equation $\mathbf{x}^T\mathbf{A} = \mathbf{0}$ was shown above to have a unique solution that is a probability vector, we have that $\mathbb{E}[\tilde{\mathsf{N}}^{(i)}(H,Q)] \longrightarrow \tilde{\mathsf{V}}(H,Q)$ as $i \to \infty$. It then follows also that the conditional expectation of $\tilde{\mathsf{N}}^{(i)}(H,Q)$ given that a begin-signal transition occurred at time 0 converges to the quantity of Equation 6.21.

Now for $t \in [0,T]$, write $\tilde{\mathsf{N}}'^{(i)}_t \in \tilde{\mathcal{B}}' \to \mathbb{R}_+$ for the stochastic process which tracks the evolution of the component counts of $\tilde{G}'^{(i)}$ after they are initialised by the state of $\tilde{G}^{(i)}$ at time 0, given

that a begin-signal transition has just occurred. Also write for the rescaled component counts $\bar{\tilde{\mathsf{N}}}_t'^{(i)} \in \mathcal{B} \to \mathbb{R}_+$ defined as follows, for $(Y,Q) \in \mathcal{B}$:

$$\bar{\tilde{\mathsf{N}}}_t'^{(i)}(Y,Q) := \begin{cases} \frac{1}{\mathcal{S}(G^{(i)})}\left(\tilde{\mathsf{N}}_t'^{(i)}(H,Q) + \sum_{R \in Q \underset{*}{\rhd\!\lhd} -} \tilde{\mathsf{N}}_t'^{(i)}(H,R)\right) & : Y = H \\ \frac{1}{\mathcal{S}(G^{(i)})}\tilde{\mathsf{N}}_t'^{(i)}(Y,Q) & : \text{otherwise} \end{cases}$$

Now by the above reasoning, we have as $i \to \infty$ that for each $(Y,Q) \in \mathcal{B}$, $\bar{\tilde{\mathsf{N}}}_0'^{(i)}(Y,Q)$ converges in probability to $\bar{\mathsf{V}}(Y,Q)$. By Corollary 6.2.3, it thus follows that $\bar{\tilde{\mathsf{N}}}_t'^{(i)}(Y,Q)$ converges in probability to $\bar{\mathsf{V}}(Y,Q)$ uniformly for all $t \in [0,T]$.

Finally, for $R \in ds^*(P \underset{*}{\rhd\!\lhd} Pb')$, Dynkin's formula gives:

$$\dot{\mathbb{E}}[\tilde{\mathsf{N}}_t'^{(i)}(H,R)] = \sum_{U \in ds^*(P \underset{*}{\rhd\!\lhd} Pb')} \mathbb{E}[D_{t,U,R}^{(i)} \times \tilde{\mathsf{N}}_t'^{(i)}(H,U)]$$

where the $D_{t,U,R}^{(i)}$ depend on $\tilde{\mathsf{N}}_t'^{(i)}$ only through the rescaled quantities $\bar{\tilde{\mathsf{N}}}_t'^{(i)}$, and, in the limit $i \to \infty$, $D_{t,U,R}^{(i)}$ converges uniformly in probability for $t \in [0,T]$ to $C_{U,R}$, given in Equation 6.23. A final application of Grönwall's inequality then completes the proof verifying that $\mathbb{E}[\tilde{\mathsf{N}}_t'^{(i)}(H,R)] \to \tilde{\mathsf{V}}_t'(H,R)$ uniformly for $t \in [0,T]$ as $i \to \infty$. □

The theorem is illustrated by Figures 4.6 and 5.11.

Recall that for a split-free model, the techniques of Sections 6.3.1 and 6.3.2 can be used to verify the condition of asymptotic stability required by this theorem. We remind the reader that although these techniques deliver only a sufficient condition for asymptotic stability, in practice we have found it to be a very powerful technique. We have not yet found any cases where we suspect asymptotic stability but this approach is not able to verify it.

In the case of splitting models, these approaches for verifying the asymptotic stability do not work. It may, in some cases, however, be possible to construct an ad-hoc Lyapunov function to verify asymptotic stability. Otherwise, experiments can be performed to build confidence empirically that a unique meaningful fixed point is asymptotically stable by testing a number of trajectories numerically for convergence to the fixed point.

6.5 Conclusion

In this chapter, we have considered key theoretical aspects of the correctness of the first-order ODE approximation. We have built additional confidence in the general analysis approach presented in this thesis by showing that the ODE approximation is exact in the limit of large component populations in the transient case (Corollary 6.2.3). This result also holds in the steady-state case (Corollary 6.3.3) subject to a condition on the stability of the approximating system of ODEs. Section 6.3.2 is devoted to developing an inexpensive and easily automated approach to verifying this condition for a large class of performance models. Finally, in Section 6.4, we develop convergence results (Theorems 6.4.1 and 6.4.2) for the passage-time approximations detailed in Chapters 4 and 5.

Chapter 7

Conclusion

7.1 Summary of achievements

This thesis has explored efficient differential-equation based techniques for the rapid extraction of useful quantitative performance measures from models of computer systems with large numbers of interacting entities. Specifically, we have focussed on models whose state-space size puts them far beyond the capabilities of traditional explicit-state analysis techniques, even when powerful symbolic representation schemes [e.g. 172] are available. As we have seen, it is not too difficult to encounter such a model — the aggregated state space of the simple client-server model $\boldsymbol{CS}(n, m)$ of Section 3.2.3 consists of up to $208,105,551$ states for the modest parameters $n = 200$ and $m = 100$.

The general ideas which motivate the approach of this thesis can be traced back to the field of statistical physics where complex Markovian models of particle interaction were approximated by so-called *mean-field models* [e.g. 26, 27, 117, 186]. More recently, these ideas have been applied in a structured manner to model computing systems in the mean-field frameworks of, for example, Le Boudec et al. [179] and Bobbio et al. [33]; and the so-called *fluid-flow* analysis approach to stochastic process algebra models [35, 147]. Our work builds directly upon the original work of Hillston [147] where differential equations approximating component counts were derived from models described in the stochastic process algebra PEPA [146]. In particular, in Chapter 3, we introduced the grouped PEPA (GPEPA) extension which removes ambiguity as to exactly when this approach can be applied at the same time as enlarging significantly the class of models to which it is applicable.

In the area of computer performance analysis, one of the key limitations of this approach up until now has been that it is restricted to the approximation of *first-order* model statistics in the form of mean component-counts. Our major contribution of Chapter 3 was to show that we may generalise this approach to allow the efficient differential-equation based approximation of arbitrary joint higher-order moments of component counts. Importantly, this allows us to obtain accurate and tractable approximations to key model statistics such as the variance of the number of components in a given state or even the skewness or kurtosis.

Although we must increase the size of the system of differential equations in order to access these additional quantities, the number of differential equations required *for any moment order*

remains *independent* of the total component population size. We have shown that first-order, second-order and very often third-order moment analysis should be eminently tractable for realistically-detailed models. Furthermore even ODE-based moment approximation beyond the third moment is often orders of magnitude more tractable than any alternative approach. We also believe that these extensions are immediately applicable to other related analysis techniques such as the mean-field approaches [e.g. 18, 19, 30, 33, 201] or continuous stochastic Petri net models [e.g. 86, 224].

Our other major contribution of Chapter 3 was to develop a *qualitative predictor* of the accuracy of the approximation which works for any order of moment analysis. In particular, by deriving the moment approximations algebraically in a formal manner from the underlying CTMC, we were able to isolate portions of the state space (so-called *switch points*) where the approximation error is likely to be at its worst. This could be used to warn a performance engineer automatically that a particular approximate analysis might not be reliable. If the engineer has the flexibility to modify the model parameters, we can even envisage automatically adjusting them in order to guarantee a more reliable analysis. Since this work was originally published in Hayden and Bradley [139], we note that Massink et al. [183, 184] have adapted our ideas to reason qualitatively about the error in a first-moment ODE-based analysis of emergent crowd behaviour. The notion of switch points also carries over directly to the context of continuous stochastic Petri net models and to the analysis of many-server queueing networks where the number of servers increases proportionately to the number of customers.

As described, the solutions of these systems of differential equations directly approximate moments of component counts, a metric which in itself can often be of direct interest to performance engineers. For example, it could be useful to ascertain the effect of varying a particular model parameter on the number of clients blocked waiting for service at a given time. However, many service level agreements and other similar quality of service guarantees are phrased in terms of *passage-time* or *response-time* quantiles, such as *a broadband connection must be successfully established within two seconds 98.5% of the time.* For this reason, Chapter 4 was devoted to exploring how various kinds of useful passage-time quantities can be extracted using an efficient ODE-based analysis approach. We identified two specific classes of passage-time query commonly asked by performance engineers: *global passage times* and *individual passage times.* We proceeded to show how such measures can be exposed explicitly in a model's state space as a *time-to-extinction* measure by a suitable local state-space transformation. This was the first step in rendering them amenable to ODE-based analysis.

Global passage times consider queries defined in terms of the joint evolution of a large population of components. For example, in a peer-to-peer file sharing model, we might ask how long it takes for half of the peers to obtain a particular file. For this performance metric, we were able to define two complementary ODE-based approximation methods. The first was a simple deterministic approximation accurate when populations are sufficiently large. In the case of non-negligible variance in the passage-time random variable of interest, we developed an approach which can be used to yield upper and lower bounds on the global passage-time's cumulative distribution function (CDF). We note that this is directly applicable since a sufficiently tight CDF lower bound can be used conservatively to validate satisfaction of a service level agreement. The bounds can be tightened by employing higher-order component-count moment analysis at the expense of a larger system of ODEs. Individual passage times are specified in terms of the time it takes for a particular individual within the system to complete some task or achieve some goal. The analogous individual passage-time query to the global query given above would

be the time taken for *one* individual peer to obtain the file. For passage-time queries of this form, we were able to derive direct ODE-defined approximations to the corresponding CDF, both for transient and steady-state measures. This work represents a direct refinement and extension of the approaches of Clark et al. [81], Ding [91], Tribastone [249] which were based on Little's Law [181]. Those authors were able to compute *mean* individual response-time results using ODE-based analysis whereas our approach extends this to the entire CDF.

After the construction of a performance model, there are two significant facets to its analysis. Computation of a response-time or other type of quantitative performance measure requires first that it can be specified accurately in a formal manner. This is made even more difficult if the behaviour of interest is not necessarily encoded explicitly in the natural state space of the model, that is, where *derived behaviour* is of interest. In Chapter 5, we developed the *unified stochastic probe* framework which builds upon the original stochastic probe definition [8], bringing together several significant enhancements. This approach allows performance measures to be described by a powerful syntax based on regular expressions which are then converted to fragments of process algebra (in this case, PEPA). These process algebra fragments are composed with the model of interest and the resulting model with an augmented state space can be analysed to extract the performance query. It is of course very important that probe fragments do not modify the behaviour of the model they are being used to query. Immediate signals are a key mechanism for allowing probes to communicate without incurring additional delay. In order to support this, we showed how immediate actions can be introduced into the GPEPA formalism, resulting in *immediate GPEPA (*i*GPEPA)*. Furthermore, we accomplished this in such a way as to ensure that ODE-based approaches are still available to us within this extended modelling framework. Finally, we showed how the tractable ODE-based passage-time approximations of Chapter 4 may be applied to compute detailed probe-specified performance measures.

Chapter 6 focused on providing theoretical justification for the validity of the first-order analyses described throughout this thesis. In the transient case, we were able to derive error bounds on the approximation, which, although extremely conservative can at least be used to derive a limiting result. Our main contribution in this chapter was the development of an analogous result for the steady-state case. We observed that the proof must be constructed in an entirely different manner to the transient case and is only guaranteed to hold when certain asymptotic stability conditions are first satisfied by the approximating system of ODEs. For a large class of GPEPA models, we developed a powerful methodology which can be used to automatically verify the required stability properties. This exact same approach could be applied directly to performance models of massively-parallel computer systems specified by continuous stochastic Petri nets or certain kinds of queueing networks since the synchronisation dynamics are very similar.[1] Finally, we were able to extend the generic convergence theorems to obtain results for the passage-time approximations detailed in Chapters 4 and 5.

7.2 Applications

The main focus of this thesis has been on providing methods for the efficient and scalable performance analysis of massively-parallel computer systems. The techniques we have described have in fact already been implemented in the modelling tool *grouped PEPA analyser (GPA)* by

[1] Specifically, the approximating system of ODEs is *piecewise affine* in both of these cases.

Stefanek et al. [229]. This tool supports the grouped PEPA syntax and allows for automatic ODE-based general moment and passage-time approximation. The tool also uses the notion of *switch points* from Chapter 3 to alert the user to potential inaccuracies in ODE approximations.

In addition to being able to perform this kind of analysis on one specific model, the GPA tool also supports the ability to perform various kinds of parameter sweeping experiments. This would allow many different model parameter combinations to be tested and potentially an optimum result established. It is in performing massive computations of this kind where the approach of this thesis becomes invaluable in reducing analysis times to a practical level. We have already used this approach to establish optimal parameter configurations which satisfy service level agreements in energy consumption models [232], cloud computing environments [233] and massively-parallel client–server systems [142].

As an example of the power afforded by the techniques presented in this thesis, we will summarise here the example customer–service system and accompanying results given in Section 5 of Hayden et al. [142]. This model is essentially a more interesting generalisation of the example client–server model of Section 3.2.3. We consider a large customer–service system with two classes of customers, service preemption and failure. The customers from each class require access to one of the services in order to complete their phase of service. Such a system, with a very large number of customers and services could for example represent a virtualised cloud-computing infrastructure, a large parallel architecture running behind an Internet search engine or a massive multimedia content provider.

We assume that the service provider offers two different levels of service level agreement (SLA) to its customers. We will express these in terms of steady-state individual passage-time measures as defined in Chapters 4 and 5. For example, high-priority customers (denoted $\boldsymbol{H}$) might be guaranteed completion within 8 seconds 95% of the time, whereas, low-priority customers (denoted $\boldsymbol{L}$) might only be guaranteed completion within 40 seconds 80% of the time. We will show how the techniques of this thesis can be used to perform efficient parameter sweeping to determine model configurations which will meet these two service requirements simultaneously. More specifically, we might be interested in computing the minimal number of active services required to ensure that both SLAs are satisfied jointly.

We now proceed to give the *i*GPEPA definition of the customer–service system. The two classes of customer are represented by the fluid components $\boldsymbol{Cust^L}$ and $\boldsymbol{Cust^H}$ whose definitions are given in Figure 7.1. Each customer sends a request to the service and waits for a response. Additionally, the high priority customers are allowed to switch to a mode where they negotiate a preemption of a low priority customer. After the response, customers wait for the service to finish, notified by the $end^{\boldsymbol{L}}$ and $end^{\boldsymbol{H}}$ actions. High priority customers can preempt the service of low priority customers, in which case the low priority service is cancelled and the customer has to restart the whole procedure. Both customers also have to cater for the possibility of the service failing and need to restart when the server has not responded for a period of time. Finally, both customers are allowed to perform their respective think action, which represents the end of the service phase for that customer.

The services are represented by the fluid components $\boldsymbol{Serv^L}$ and $\boldsymbol{Serv^H}$ whose definitions are given in Figure 7.2. The services alternate between listening to requests from low- or high-priority customers. After a request, the services initiate the processing, with a possibility of failing. In addition, services serving low priority customers listen to requests for preemption, which cause the service to drop the served customer and switch to a higher priority customer.

$$
\begin{aligned}
\boldsymbol{Cust^L} &\stackrel{def}{=} (request^L, r_{request}).\boldsymbol{Cust^L_{proc}} \\
\boldsymbol{Cust^L_{proc}} &\stackrel{def}{=} (end^L, r_{end}).\boldsymbol{Cust^L_{think}} + (drop, r_{drop}).\boldsymbol{Cust^L} + (timeout, r_t^B).\boldsymbol{Cust^L} \\
\boldsymbol{Cust^L_{think}} &\stackrel{def}{=} (think, r_{think}).\boldsymbol{Cust^L} \\
\boldsymbol{Cust^H} &\stackrel{def}{=} (request^H, r_{request}).\boldsymbol{Cust^H_{proc}} + (timeout, r_t^H).\boldsymbol{Cust^H_{preempt}} \\
\boldsymbol{Cust^H_{proc}} &\stackrel{def}{=} (end^H, r_{end}).\boldsymbol{Cust^H_{think}} + (timeout, r_t^B).\boldsymbol{Cust^H} \\
\boldsymbol{Cust^H_{preempt}} &\stackrel{def}{=} (preempt, r_{preempt}).\boldsymbol{Cust^H_{proc}} + (timeout, r_t^H).\boldsymbol{Cust^H} \\
\boldsymbol{Cust^H_{think}} &\stackrel{def}{=} (think, r_{think}).\boldsymbol{Cust^H}
\end{aligned}
$$

Figure 7.1: Fluid components $\boldsymbol{Cust^L}$ and $\boldsymbol{Cust^H}$.

$$
\begin{aligned}
\boldsymbol{Serv^L} &\stackrel{def}{=} (request^L, r_{request}).\boldsymbol{Serv^L_{proc}} + (switch, r^{LH}_{switch}).\boldsymbol{Serv^H} \\
\boldsymbol{Serv^L_{proc}} &\stackrel{def}{=} (proc, r^L_{proc}).\boldsymbol{Serv^L_{end}} + (preempt, r_{preempt}).\boldsymbol{Serv^L_{drop}} + (break, r_{break}).\boldsymbol{Serv^L} \\
\boldsymbol{Serv^L_{drop}} &\stackrel{def}{=} (drop, r_{drop}).\boldsymbol{Serv^H_{proc}} + (break, r_{break}).\boldsymbol{Serv^L} \\
\boldsymbol{Serv^H} &\stackrel{def}{=} (request^H, r_{request}).\boldsymbol{Serv^H_{proc}} + (switch, r^{HL}_{switch}).\boldsymbol{Serv^L} \\
\boldsymbol{Serv^H_{proc}} &\stackrel{def}{=} (proc, r^H_{proc}).\boldsymbol{Serv^H_{end}} + (break, r_{break}).\boldsymbol{Serv^H}
\end{aligned}
$$

Figure 7.2: Fluid components $\boldsymbol{Serv^L}$ and $\boldsymbol{Serv^H}$.

Finally, when finished processing, the servers notify the respective customers.

The complete iGPEPA model consists of n_C^L copies of the component $\boldsymbol{Cust^L}$ and n_C^H copies of $\boldsymbol{Cust^H}$. These are composed together with n_S^L services initially in state $\boldsymbol{Serv^L}$ and n_S^H initially in state $\boldsymbol{Serv^H}$:

$$
\begin{aligned}
&\boldsymbol{CustServ}(n_C^L, n_C^H, n_S^L, n_S^H) \stackrel{def}{=} \\
&\quad \Big(\mathbf{Custs^L}\{\boldsymbol{Cust^L}[n_C^L]\} \parallel \mathbf{Custs^H}\{\boldsymbol{Cust^H}[n_C^H]\}\Big) \underset{L}{\bowtie} \mathbf{Servs}\{\boldsymbol{Serv^L}[n_S^L] \wr\!\wr \boldsymbol{Serv^H}[n_S^H]\}
\end{aligned}
$$

where the cooperation set is given by $L := \{request^L, request^H, end^L, end^H, preempt, drop\}$. The full systems of first- and second-moment ODEs for the model $\boldsymbol{CustServ}(n_C^L, n_C^H, n_S^L, n_S^H)$ according to Definition 3.3.1 are available online [131].

The steady-state individual passage-time measurements forming the basis of the SLAs are given in the language of unified stochastic probes by $\boldsymbol{PM}^L$ for the low-priority customers:

$$
\begin{aligned}
\boldsymbol{PM}^L \stackrel{def}{=}\ & \mathsf{begin} : \mathsf{start}, \mathsf{end} : \mathsf{stop}^{\hookleftarrow} \qquad (7.1)\\
& \mathsf{observes} \quad \boldsymbol{Probe}_l \stackrel{def}{=} \epsilon : \mathsf{begin}, think : \mathsf{end}^{\hookleftarrow} \\
& \mathsf{where} \quad \mathbf{Custs^L}\{\boldsymbol{Cust^L}[?n]\} \Longrightarrow \mathbf{Custs^L}\{(\boldsymbol{Cust^L} \underset{*}{\bowtie} \boldsymbol{Probe}_l) \wr\!\wr \boldsymbol{Cust^L}[?n-1]\} \\
& \mathsf{in} \quad \boldsymbol{CustServ}(n_C^L, n_C^H, n_S^L, n_S^H)
\end{aligned}
$$

and $\boldsymbol{PM}^H$ for the high-priority customers:

$$\begin{aligned}
\boldsymbol{PM}^H \stackrel{def}{=}\ & \mathsf{begin} : \mathsf{start}, \mathsf{end} : \mathsf{stop}^{\hookleftarrow} \\
& \mathsf{observes} \quad \boldsymbol{Probe}_l \stackrel{def}{=} \epsilon : \mathsf{begin}, \mathit{think} : \mathsf{end}^{\hookleftarrow} \\
& \mathsf{where} \quad \mathbf{Custs^H}\{\boldsymbol{Cust}^{\boldsymbol{H}}[?n]\} \Longrightarrow \mathbf{Custs^H}\{(\boldsymbol{Cust}^{\boldsymbol{H}} \underset{*}{\bowtie} \boldsymbol{Probe}_l) \wr\wr \boldsymbol{Cust}^{\boldsymbol{H}}[?n-1]\} \\
& \mathsf{in} \quad \boldsymbol{CustServ}(n_C^L, n_C^H, n_S^L, n_S^H)
\end{aligned} \tag{7.2}$$

Following Section 5.5.1, to compute the ODE approximation to each steady-state individual passage time, ODEs must be constructed according to Definition 3.3.1 for the probed versions of the models. In particular, for the low-priority SLA, we must consider the models:

$$\begin{aligned}
\widetilde{\boldsymbol{CustServ}}_{\boldsymbol{L}}(n_C^L, n_C^H, n_S^L, n_S^H) \stackrel{def}{=} & \\
\Big(\mathbf{Custs^L}\{(\boldsymbol{Cust}^{\boldsymbol{L}} \underset{\{think\}}{\bowtie} \boldsymbol{Pb}_S) \wr\wr \boldsymbol{Cust}^{\boldsymbol{L}}[n_C^L - 1]\} \parallel \mathbf{Custs^H}\{\boldsymbol{Cust}^{\boldsymbol{H}}[n_C^H]\}\Big) & \\
\underset{L}{\bowtie} \mathbf{Servs}\{\boldsymbol{Serv}^{\boldsymbol{L}}[n_S^L] \wr\wr \boldsymbol{Serv}^{\boldsymbol{H}}[n_S^H]\} &
\end{aligned}$$

and:

$$\begin{aligned}
\widetilde{\boldsymbol{CustServ}}'_{\boldsymbol{L}}(n_C^L, n_C^H, n_S^L, n_S^H) \stackrel{def}{=} & \\
\Big(\mathbf{Custs^L}\{(\boldsymbol{Cust}^{\boldsymbol{L}} \underset{\{think\}}{\bowtie} \boldsymbol{Pb}'_S) \wr\wr \boldsymbol{Cust}^{\boldsymbol{L}}[n_C^L - 1]\} \parallel \mathbf{Custs^H}\{\boldsymbol{Cust}^{\boldsymbol{H}}[n_C^H]\}\Big) & \\
\underset{L}{\bowtie} \mathbf{Servs}\{\boldsymbol{Serv}^{\boldsymbol{L}}[n_S^L] \wr\wr \boldsymbol{Serv}^{\boldsymbol{H}}[n_S^H]\} &
\end{aligned}$$

where the repeating local probe $\epsilon : \mathsf{begin}, \mathit{think} : \mathsf{end}^{\hookleftarrow}$ is translated to the iPEPA fluid component $\boldsymbol{Pb}_S$ (following Section 5.3.2) and the absorbing version $\epsilon : \mathsf{begin}, \mathit{think} : \mathsf{end}$ is translated to the iPEPA fluid component $\boldsymbol{Pb}'_S$ as given in Figure 7.3. For the high-priority SLA, we must consider the analogous models:

$$\begin{aligned}
\widetilde{\boldsymbol{CustServ}}_{\boldsymbol{H}}(n_C^L, n_C^H, n_S^L, n_S^H) \stackrel{def}{=} & \\
\Big(\mathbf{Custs^L}\{\boldsymbol{Cust}^{\boldsymbol{L}}[n_C^L]\} \parallel \mathbf{Custs^H}\{(\boldsymbol{Cust}^{\boldsymbol{H}} \underset{\{think\}}{\bowtie} \boldsymbol{Pb}_S) \wr\wr \boldsymbol{Cust}^{\boldsymbol{H}}[n_C^H - 1]\}\Big) & \\
\underset{L}{\bowtie} \mathbf{Servs}\{\boldsymbol{Serv}^{\boldsymbol{L}}[n_S^L] \wr\wr \boldsymbol{Serv}^{\boldsymbol{H}}[n_S^H]\} &
\end{aligned}$$

and:

$$\begin{aligned}
\widetilde{\boldsymbol{CustServ}}'_{\boldsymbol{H}}(n_C^L, n_C^H, n_S^L, n_S^H) \stackrel{def}{=} & \\
\Big(\mathbf{Custs^L}\{\boldsymbol{Cust}^{\boldsymbol{L}}[n_C^L]\} \parallel \mathbf{Custs^H}\{(\boldsymbol{Cust}^{\boldsymbol{H}} \underset{\{think\}}{\bowtie} \boldsymbol{Pb}'_S) \wr\wr \boldsymbol{Cust}^{\boldsymbol{H}}[n_C^H - 1]\}\Big) & \\
\underset{L}{\bowtie} \mathbf{Servs}\{\boldsymbol{Serv}^{\boldsymbol{L}}[n_S^L] \wr\wr \boldsymbol{Serv}^{\boldsymbol{H}}[n_S^H]\} &
\end{aligned}$$

After vanishing state removal (as in Section 5.2.1), we construct 12 first-moment ODEs associated to the models $\widetilde{\boldsymbol{CustServ}}_{\boldsymbol{L}}$ and $\widetilde{\boldsymbol{CustServ}}_{\boldsymbol{H}}$, 15 to the model $\widetilde{\boldsymbol{CustServ}}'_{\boldsymbol{L}}$ and 16 to the model $\widetilde{\boldsymbol{CustServ}}'_{\boldsymbol{H}}$. These ODEs can then be used as detailed in Section 5.5.1 to compute approximations to the CDFs of interest. Figure 7.4 shows an example computation of the CDFs for the low- and high-prority customers with example SLAs for each. We see that for this

$$\boldsymbol{Pb}_S \stackrel{def}{=} \mathsf{begin}.\boldsymbol{Pb}_F$$
$$\boldsymbol{Pb}_F \stackrel{def}{=} (think, \top).\boldsymbol{Pb}_{Ft}$$
$$\boldsymbol{Pb}_{Ft} \stackrel{def}{=} \mathsf{end}.\boldsymbol{Pb}_S$$

$$\boldsymbol{Pb}'_S \stackrel{def}{=} \mathsf{begin}.\boldsymbol{Pb}'_F$$
$$\boldsymbol{Pb}'_F \stackrel{def}{=} (think, \top).\boldsymbol{Pb}'_{Ft}$$
$$\boldsymbol{Pb}'_{Ft} \stackrel{def}{=} \mathsf{end}.\boldsymbol{Pb}_{\mathrm{Acc}}$$
$$\boldsymbol{Pb}_{\mathrm{Acc}} \stackrel{def}{=} (think, \top).\boldsymbol{Pb}_{\mathrm{Acc}}$$

Figure 7.3: Translation of the local probe in the measure specifications of Equations 7.1 and 7.2 and its absorbing version to *i*PEPA components.

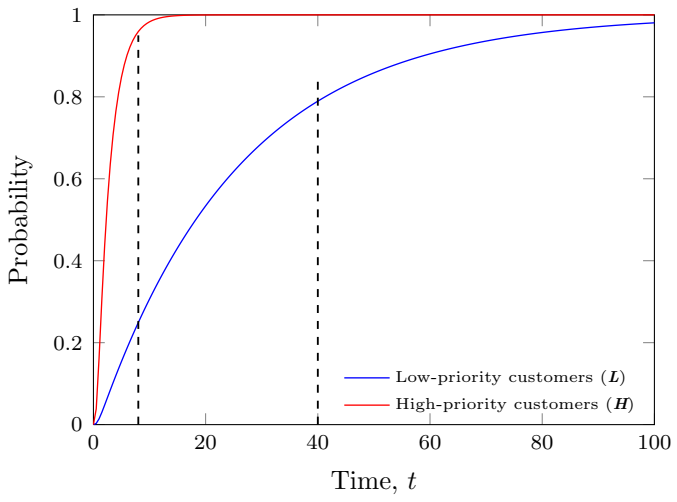


Figure 7.4: ODE approximations of the passage-time CDFs for the steady-state individual passage times specified by Equations 7.1 and 7.2. The dashed lines correspond to the respective SLAs, in this case, 8 seconds 95% of the time for the high-priority customers, and 40 seconds 85% of the time for the low-priority customers. Rates: $r_{request} = 2.0, r_t^H = 1.0, r_{preempt} = 3.0, r_{proc}^L = 1.0, r_{proc}^H = 2.0, r_{break} = 1.0, r_t^B = 1.0, r_{end} = 2.0, r_{switch}^{LH} = 1.0, r_{switch}^{HL} = 1.8, r_{think} = 3.0$. Initial component counts: $n_C^L = 100$, $n_C^H = 30$, $n_S^L = 20$, $n_S^H = 20$.

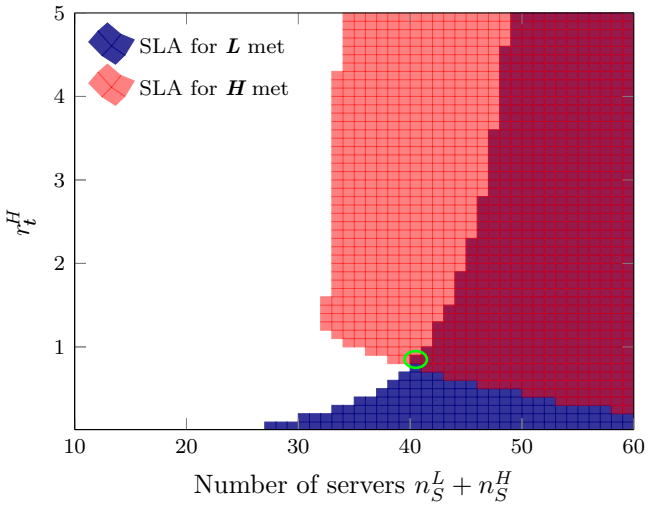


Figure 7.5: System configurations satisfying SLAs (8 seconds 95% of the time for the high-priority customers, and 40 seconds 80% of the time for the low-priority customers) over the parameter space of $n_S^L + n_S^H$ and r_t^H. The circled configuration is that which minimises the number of required services but still satisfies both SLAs. Rates: $r_{request} = 2.0, r_{preempt} = 3.0, r_{proc}^L = 1.0, r_{proc}^H = 2.0, r_{break} = 1.0, r_t^B = 1.0, r_{end} = 2.0, r_{switch}^{LH} = 1.0, r_{switch}^{HL} = 1.8, r_{think} = 3.0$. Initial component counts: $n_C^L = 100$, $n_C^H = 30$.

particular model configuration, the SLA is satisfied in the case of the high-prority customers but not for the low-priority customers.

Since the model $\boldsymbol{CustServ}(n_C^L, n_C^H, n_S^L, n_S^H)$ is split-free, we may apply the techniques of Sections 6.3.1 and 6.3.2 aimed at verifying asymptotic stability of the fixed point of the associated system of ODEs. Indeed, in the parameter regime of Figure 7.4, the necessary system of 64 linear matrix inequalities can be constructed and a feasible solution found in only a few seconds on a standard desktop computer. Theorem 6.4.2 then applies and can be used to verify that the ODE approximations to the CDFs will converge as the component populations are scaled up.

As mentioned earlier, the service provider is probably interested in satisfying both SLAs with the minimal number of running services. That is, they wish to minimise the quantity $n_S^L + n_S^H$.[2] They may also have control over other model parameters such as the rate at which high-priority customers are allowed to preempt the service of low-priority customers r_t^H. Due to the extremely efficient nature of the ODE approximation techniques, brute-force parameter sweeping over both parameters $n_S^L + n_S^H$ and r_t^H is a realistic means of determining the minimum number of services required to satisfy both SLAs. Figure 7.5 shows the results of one such parameter sweep experiment consisting of 2,500 different parameters configurations which takes only around a minute to perform on a standard desktop computer.

[2]Since both types of service alternate between serving low- and high-priority customers, steady-state measures depend only on the initial sum $n_S^L + n_S^H$.

7.3 Future work

In this section we will outline some key areas for future development, some of which have already been tentatively explored in publications outside the immediate scope of this thesis.

7.3.1 Accuracy of the ODE-based analysis approach

There are many possible ways in which the techniques developed in this thesis could be improved upon in terms of the accuracy of the approximations. Increasing the strength of theoretical guarantee that we are able to make regarding their accuracy is also of massive interest so that we can improve the reliability of the approach.

One avenue for investigation is the possibility to derive *stochastic differential equations (SDEs)* from performance models of massively-parallel systems. We have already made preliminary progress in this area [138, 229] and have shown how SDEs can be derived from PEPA models whose solutions quantify the deviation of the underlying CTMC from the approximating ODE. This work suggests that the distribution of component counts can be approximated at the second order by the multivariate normal distribution whose parameters are obtained by solving the first- and second-moment ODEs of Chapter 3. Such an approximation is of course more detailed and provides more information than the first- and second-moment information alone.

ODE-based moment approximation techniques similar to those developed in this thesis for GPEPA have also been developed for the analysis of chemical and biological models [e.g. 110, 180]. Furthermore, various schemes known collectively as *moment closure* methods have been developed to allow higher-order moment information to be fed back into improving the approximation at lower-orders. For example, the variance information obtained by solving the second-order moment ODEs might be used to improve the first-order approximation. A similar extension in the context of GPEPA constitutes potential for future work. This is not immediately straightforward due to the non-smooth nature of the synchronisation dynamics common to models of computer and communication systems.[3]

The discussion in Chapter 6 regarding the extremely conservative general nature of the error bounds given for the first-order ODE approximation by Theorems 6.2.1 and 6.2.2 suggests a potentially fruitful future research avenue. In order to improve upon these bounds in any significant manner, it is clearly necessary to dispense with the crude device of Grönwall's inequality. Instead, it would seem massive improvements might be had by a more careful consideration of the stability of the approximating system of ODEs' trajectories. A starting point for such an investigation might be the recent work of Turner [254] where martingales are constructed from the linearisation of a differential equation approximating a simple Markov chain. This appears to allow local stability information for trajectories to be considered directly in the construction of error bounds.

[3] In for example, GPEPA and continuous stochastic Petri nets, this comes from the presence of minimum terms in the CTMC transition rates.

7.3.2 Capturing generally-timed transitions with DDEs

Generally-timed delays destroy the Markovian nature of a model making it even more difficult to analyse. Even so, they are key if realistic models are desired. Ordinarily, such distributions are incorporated into a Markovian model by using an approximating phase-type distribution [e.g. 10]. However, many phases are often required for an accurate approximation so this can result in costly growth of the discrete state-space of local components. In turn, this results in a growth in the approximating system of ODEs if the techniques of this thesis are to be applied. This is especially true if higher-order moment analysis is to be applied.

We wish to investigate the use of recently-developed *delay differential equation (DDE)* [215] approximation techniques to capture these features in a scalable fashion. DDEs are already a popular modelling tool for biological systems [24, 215]. We have completed initial work exploring such possibilities in Hayden [132].

7.3.3 More efficient global optimisation

As we have illustrated earlier in this chapter, the massive reduction in analysis time offered by the ODE-based techniques of this thesis have already opened the door to performing optimisation and scalability analysis over many parameters simultaneously in a practical length of time [142, 232, 233]. However, the inevitable growth in the dimensionality of parameter spaces we wish to consider represents an orthogonal exponential growth, a kind of *parameter space explosion*.

In order to address this, we intend to consider more efficient approaches to the global optimisation problem in our specific context of a tractable representation by differential equations. Specifically, we wish to explore techniques from the field of *verified numerical integration* such as *interval methods* and and *Taylor-model arithmetic* [195, 198, 213, 226]. We have completed initial work exploring such possibilities in Hayden [133].

7.3.4 Multi-stage synchronisation in massively-parallel systems

A key limitation of PEPA (and also grouped PEPA) is its inability to encode the idea of a multi-stage *transaction* between two or more cooperating components. To illustrate what we mean by this, recall the client–server model from Section 3.2.3, which we repeat here in Figure 7.6.

Clients first make a request to one of the m available servers (the *request*-action), after which the intention of the modeller is probably supposed to be that they receive data from that *same* server (the *data*-action). However, due to the semantics of PEPA, there is no guarantee that the *data*-action will be executed in cooperation with the *same* server component with whom the *request*-action was originally completed. The GPEPA process algebra fails to capture the idea of a *transaction* between a given client and a given server. That is to say, there is no way to enforce that when a *request*-action happens, a binding is established between the client and the server which lasts for the entire duration of their interaction. The only way to encode this behaviour here would be to define many different *request*- and *data*-actions to capture explicitly the possible pairings of clients and servers. This is clearly impractical and directly breaks the compositional structure of the model.

$$\begin{aligned}
\boldsymbol{Client} &\stackrel{def}{=} (request, r_r).\boldsymbol{Client_{waiting}} \\
\boldsymbol{Client_{waiting}} &\stackrel{def}{=} (data, r_d).\boldsymbol{Client_{think}} + (timeout, r_{tmt}).\boldsymbol{Client} \\
\boldsymbol{Client_{think}} &\stackrel{def}{=} (think, r_t).\boldsymbol{Client} \\
\boldsymbol{Server} &\stackrel{def}{=} (request, r_r).\boldsymbol{Server_{get}} + (break, r_b).\boldsymbol{Server_{broken}} \\
\boldsymbol{Server_{get}} &\stackrel{def}{=} (data, r_d).\boldsymbol{Server} + (break, r_b).\boldsymbol{Server_{broken}} \\
\boldsymbol{Server_{broken}} &\stackrel{def}{=} (reset, r_{rst}).\boldsymbol{Server} \\
\\
\boldsymbol{CS}(n, m) &\stackrel{def}{=} \mathbf{Clients}\{\boldsymbol{Client}[n]\} \underset{\{request, data\}}{\bowtie} \mathbf{Servers}\{\boldsymbol{Server}[m]\}
\end{aligned}$$

Figure 7.6: *i*GPEPA description of the simple client–server model $\boldsymbol{CS}(n, m)$.

$$\begin{aligned}
\boldsymbol{Client}^* &\stackrel{def}{=} (request, {r_r}^*).\boldsymbol{Client}^*_{\boldsymbol{waiting}} \\
\boldsymbol{Client}^*_{\boldsymbol{waiting}} &\stackrel{def}{=} (data, {r_d}^*).\boldsymbol{Client}^*_{\boldsymbol{think}} + (timeout, {r_{tmt}}^*).\boldsymbol{Client} \\
\boldsymbol{Client}^*_{\boldsymbol{think}} &\stackrel{def}{=} (think, {r_t}^*).\boldsymbol{Client}^* \\
\boldsymbol{Server}^* &\stackrel{def}{=} (request, {r_r}^*).\boldsymbol{Server}^*_{\boldsymbol{get}} + (break, {r_b}^*).\boldsymbol{Server}^*_{\boldsymbol{broken}} \\
\boldsymbol{Server}^*_{\boldsymbol{get}} &\stackrel{def}{=} (data, {r_d}^*).\boldsymbol{Server}^* + (break, {r_b}^*).\boldsymbol{Server}^*_{\boldsymbol{broken}} \\
\boldsymbol{Server}^*_{\boldsymbol{broken}} &\stackrel{def}{=} (reset, {r_{rst}}^*).\boldsymbol{Server}^* \\
\\
\boldsymbol{CS}^*(n, o, m, p) &\stackrel{def}{=} \mathbf{Clients}\{\boldsymbol{Client}[n] \wr\!\wr \boldsymbol{Client}^*[o]\} \underset{L}{\bowtie} \mathbf{Servers}\{\boldsymbol{Server}[m] \wr\!\wr \boldsymbol{Server}^*[p]\}
\end{aligned}$$

Figure 7.7: *i*GPEPA description of client–server model with two classes of client and server $\boldsymbol{CS}^*(n, o, m, p)$.

Fortunately, in terms of the aggregate behaviour of component counts, this modelling inaccuracy makes no difference to the analysis for many very simple models — indeed, this is the case for the model $\boldsymbol{CS}(n, m)$. If we are only interested in aggregate component-count behaviour, it does not matter if a client performs the *request*-action and the *data*-action with different servers because the behaviour of each server is identical anyway. However, if we were to extend this model by introducing multiple different classes of client and server with different behaviour, this is no longer necessarily true. To see this, imagine adding an additional class of client, say $\boldsymbol{Client}^*$, and of server, say $\boldsymbol{Server}^*$, to the client–server model $\boldsymbol{CS}(n, m)$, as in Figure 7.7.

Assume that the rates for these new fluid components are different to those for the original client and server components thus giving them different timed behaviour. If a particular type of client begins an interaction with a particular type of server, this binding between component classes should be maintained for the duration of the interaction in order to capture a realistic scenario. Otherwise, a client is able to switch between server classes part way through a shared interaction. Then it is possible for the client to experience what should be an impossible interaction consisting of a combination of parts of the timed behaviour of the $\boldsymbol{Server}$ component

$$\begin{aligned}
\boldsymbol{Client} &\stackrel{def}{=} (request\ \mathsf{get}\ x, r_r).\boldsymbol{Client_{waiting}}(x)\\
\boldsymbol{Client_{waiting}}(x) &\stackrel{def}{=} (data\ \mathsf{on}\ x, \top).\boldsymbol{Client_{think}} + (timeout, r_{tmt}).\boldsymbol{Client}\\
\boldsymbol{Client_{think}} &\stackrel{def}{=} (think, r_t).\boldsymbol{Client}\\
\boldsymbol{Server} &\stackrel{def}{=} (request\ \mathsf{init}\ x, r_r).\boldsymbol{Server_{get}}(x) + (break, r_b).\boldsymbol{Server_{broken}}\\
\boldsymbol{Server_{get}}(x) &\stackrel{def}{=} (data\ \mathsf{on}\ x, r_d).\boldsymbol{Server} + (break, r_b).\boldsymbol{Server_{broken}}\\
\boldsymbol{Server_{broken}} &\stackrel{def}{=} (reset, r_{rst}).\boldsymbol{Server}
\end{aligned}$$

Figure 7.8: *i*GPEPA description of client and server fluid components with channels.

and the $\boldsymbol{Server}^*$ component.

In order to address these issues, we have recently begun work on *grouped PEPA with channels* [233] which adds a mechanism for maintaining links between components currently involved in a shared transaction. This builds on our earlier work in defining *shared transaction Markov chains (STMCs)* [137], a low-level performance modelling formalism supporting transactions between cooperating components.

Our use of channels is similar to that of π-calculus [188], although instead of having single named *input* and *output* prefixes for communication across channels, we utilise the existing PEPA cooperation framework thus supporting also many-way synchronisation across channels between transaction participants. Within this framework, we may rewrite the fluid components $\boldsymbol{Client}$ and $\boldsymbol{Server}$ in the model $\boldsymbol{CS}(n, m)$ as in Figure 7.8.

The *channel variable* x is used to make the formal binding between a particular instance of a client and server component during a shared interaction. Specifically, the syntax *data* on x means that the *data*-action can only occur between $\boldsymbol{Client_{waiting}}$ and $\boldsymbol{Server_{get}}$ components who already share the same channel in x. The syntax *request* init x means that the $\boldsymbol{Server}$ component creates a new unique channel instance which is then shared with a $\boldsymbol{Client}$ component (by virtue of the *request* get x syntax) when they first complete a *request*-action together. The extension also supports *forwarding* of channels to support nested layering of services.

At present, this work is still very much at the proof of concept stage and there are many avenues for further research. In particular, a formal definition of the operational semantics of the channel extension is still outstanding. Although we have provided ad-hoc derivations of approximating systems of ODEs in Stefanek et al. [233], this has also yet to be considered more generally for GPEPA models with channels.

Furthermore, there are many systems for which the operational semantics of PEPA are not appropriate since they do not capture the synchronisation or scheduling policies of the system accurately. We are thus also interested in adapting the ODE-based techniques of this thesis to a more general formalism where a separation between interaction and timing is maintained as in, for example, *interactive Markov chains (IMC)* [143] or *shared transaction Markov chains (STMCs)* [137]. In this type of formalism, synchronisation events are untimed and happen immediately which allows for a much richer set of possible interaction mechanisms between components to be captured. On the other hand, however, as has already been touched on

by Hayden and Bradley [137], ODE-based techniques cannot always be directly extended to such richer modelling formalisms.

Specifically, mapping synchronisation onto immediate actions can result in indicator functions appearing in the right-hand of the approximating system of ODEs. Such systems are not guaranteed to have a solution when interpreted as ordinary differential equations. One approach we are investigating is to instead consider an approximating *differential inclusion* as motivated by the recent work of Gast and Gaujal [106].

Bibliography

[1] Amazon Elastic Compute Cloud (Amazon EC2), March 2011. URL `http://aws.amazon.com/ec2/`. 1

[2] GoGrid, March 2011. URL `http://www.gogrid.com/`. 1

[3] Google App Engine, March 2011. URL `http://code.google.com/appengine/`. 1

[4] V. Acary and B. Brogliato. *Numerical methods for nonsmooth dynamical systems: applications in mechanics and electronics*. Springer, 2008. 28

[5] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. Modelling with Generalized Stochastic Petri Nets. *ACM SIGMETRICS Performance Evaluation Review*, 26(2):2, Aug. 1998. ISSN 01635999. doi: 10.1145/288197.581193. URL `http://portal.acm.org/citation.cfm?id=288197.581193`. 77

[6] E. Altman, P. Nain, A. Shwartz, and X. Yuedong. Predicting the Impact of Measures Against P2P Networks on the Transient Behaviors. In *30th IEEE International Conference on Computer Communications (IEEE INFOCOM)*, April 2011. 28

[7] H. Andersson and T. Britton. *Stochastic epidemic models and their statistical analysis*. Springer, 2000. 4

[8] A. Argent-Katwala, J. T. Bradley, and N. J. Dingle. Expressing performance requirements using regular expressions to specify stochastic probes over process algebra models. In *Proceedings of the 4th International Workshop on Software and Performance*, pages 49–58, May 2004. URL `http://en.scientificcommons.org/43251272`. 32, 76, 77, 82, 142

[9] A. Argent-Katwala, J. T. Bradley, A. Clark, and S. T. Gilmore. Location-Aware Quality of Service Measurements for Service-Level Agreements. In G. Barthe and C. Fournet, editors, *Trustworthy Global Computing*, volume 4912 of *Lecture Notes in Computer Science*, pages 222–239, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-78662-7. doi: 10.1007/978-3-540-78663-4. URL `http://www.springerlink.com/content/ug3u180368820548/`. 77, 78, 82

[10] S. Asmussen, O. Nerman, and M. Olsson. Fitting Phase-Type Distributions via the EM Algorithm. *Scandinavian Journal of Statistics*, 23(4):419–441, Nov. 1996. URL `http://www.jstor.org/pss/4616418`. 149

[11] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Verifying continuous time Markov chains. In R. Alur and T. Henzinger, editors, *Computer Aided Verification*, volume 1102 of *Lecture Notes in Computer Science*, pages 269–276. Springer Berlin / Heidelberg,

1996. doi: 10.1007/3-540-61474-5_75. URL http://www.springerlink.com/content/y41217212k6329q7/. 76

[12] F. Baccelli, D. McDonald, and J. Reynier. A mean-field model for multiple TCP connections through a buffer implementing RED. *Performance Evaluation*, 49(1-4):77–97, Sept. 2002. ISSN 01665316. doi: 10.1016/S0166-5316(02)00136-0. URL http://dx.doi.org/10.1016/S0166-5316(02)00136-0. 4, 27

[13] F. Baccelli, A. Chaintreau, D. De Vleeschauwer, and D. McDonald. HTTP Turbulence. *Networks and Heterogeneous Media*, 1(1):1–40, 2006. 4, 27

[14] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time markov chains. *IEEE Transactions on Software Engineering*, 29(6):524–541, June 2003. ISSN 0098-5589. doi: 10.1109/TSE.2003.1205180. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1205180. 58

[15] C. Baier, L. Cloth, B. R. Haverkort, M. Kuntz, and M. Siegle. Model Checking Markov Chains with Actions and State Labels. *IEEE Transactions on Software Engineering*, 33(4):209–224, Apr. 2007. ISSN 0098-5589. doi: 10.1109/TSE.2007.36. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4123324. 76, 82, 113

[16] R. Bakhshi. *Gossiping Models*. Ph. D., Vrije Universiteit Amsterdam, January 2011. URL http://www.few.vu.nl/~rbakhshi/papers/DARE-color.pdf. 27

[17] R. Bakhshi, L. Cloth, W. Fokkink, and B. R. Haverkort. Mean-Field Analysis for the Evaluation of Gossip Protocols. In *Sixth International Conference on the Quantitative Evaluation of Systems*, pages 247–256. IEEE, Sept. 2009. ISBN 978-0-7695-3808-2. doi: 10.1109/QEST.2009.38. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5290827. 27

[18] R. Bakhshi, J. Endrullis, S. Endrullis, W. Fokkink, and B. R. Haverkort. Automating the Mean-Field Method for Large Dynamic Gossip Networks. In *Seventh International Conference on the Quantitative Evaluation of Systems*, pages 241–250. IEEE, Sept. 2010. ISBN 10.1109/QEST.2010.38. URL http://www.computer.org/portal/web/csdl/doi/10.1109/QEST.2010.38. 4, 27, 30, 141

[19] R. Bakhshi, L. Cloth, W. Fokkink, and B. R. Haverkort. Mean-field framework for performance evaluation of push-pull gossip protocols. *Performance Evaluation*, 68(2):157–179, Feb. 2011. ISSN 01665316. doi: 10.1016/j.peva.2010.08.025. URL http://dx.doi.org/10.1016/j.peva.2010.08.025. 27, 30, 141

[20] G. Balbo. Introduction to Stochastic Petri Nets. In E. Brinksma, H. Hermanns, and J.-P. Katoen, editors, *Lectures on Formal Methods and Performance Analysis*, volume 2090 of *Lecture Notes in Computer Science*, pages 84–155. Springer Berlin Heidelberg, Sept. 2001. ISBN 978-3-540-42479-6. doi: 10.1007/3-540-44667-2. URL http://www.springerlink.com/content/0dg47hmt2ynjyk2m/. 29, 77

[21] G. Balbo, M. Beccuti, M. De Pierro, and G. Franceschinis. First Passage Time Computation in Tagged GSPNs with Queue Places. *The Computer Journal*, July 2010. ISSN 0010-4620. doi: 10.1093/comjnl/bxq056. URL http://comjnl.oxfordjournals.org/cgi/content/abstract/bxq056v1. 58

[22] P. Ballarini. *Towards compositional CSL model-checking.* Ph. D., Università di Torino, 2004. 18

[23] P. Ballarini and A. Horváth. Compositional Model Checking of Product-Form CTMCs. In *Seventh International Workshop on Automated Verification of Critical Systems (AVoCS)*, volume 250 of *Electronic Notes in Theoretical Computer Science*, pages 21–37, Sept. 2007. doi: 10.1016/j.entcs.2009.08.003. URL `http://dx.doi.org/10.1016/j.entcs.2009.08.003`. 18

[24] R. Barbuti, G. Caravagna, P. Milazzo, and A. Maggiolo-Schettini. On the Interpretation of Delays in Delay Stochastic Simulation of Biological Systems. In *2nd International Workshop on Computational Models for Cell Processes (COMPMOD)*, volume 6 of *Electronic Proceedings in Theoretical Computer Science*, pages 17–29, Oct. 2009. doi: 10.4204/EPTCS.6.2. URL `http://arxiv.org/abs/0910.1219`. 149

[25] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios. Open, Closed, and Mixed Networks of Queues with Different Classes of Customers. *Journal of the ACM*, 22(2): 248–260, Apr. 1975. ISSN 00045411. doi: 10.1145/321879.321887. URL `http://portal.acm.org/citation.cfm?id=321879.321887`. 18

[26] R. J. Baxter. *Exactly Solved Models in Statistical Mechanics.* Academic Press, 1982. 4, 27, 140

[27] N. Bellomo and M. Pulvirenti. *Modeling in applied sciences - A kinetic theory approach.* Springer, 2000. 4, 27, 140

[28] A. Ben-Israel and T. N. E. Greville. *Generalized Inverses: Theory and Applications.* Springer, second edition, 2003. 123

[29] M. Benaïm. Recursive algorithms, urn processes and chaining number of chain recurrent sets. *Ergodic Theory and Dynamical Systems*, 18(01):53–87, Feb. 1998. ISSN 1469-4417. URL `http://journals.cambridge.org/abstract_S0143385798097557`. 5, 27, 128

[30] M. Benaïm and J.-Y. Le Boudec. A class of mean field interaction models for computer and communication systems. *Performance Evaluation*, 65(11-12):823–838, Nov. 2008. ISSN 01665316. doi: 10.1016/j.peva.2008.03.005. URL `http://dx.doi.org/10.1016/j.peva.2008.03.005`. 27, 30, 128, 141

[31] M. Bernardo. A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time. *Theoretical Computer Science*, 202(1-2): 1–54, July 1998. ISSN 03043975. doi: 10.1016/S0304-3975(97)00127-8. URL `http://dx.doi.org/10.1016/S0304-3975(97)00127-8`. 77

[32] P. Billingsley. *Convergence of Probability Measures.* John Wiley & Sons, 1968. 129

[33] A. Bobbio, M. Gribaudo, and M. Telek. Analysis of Large Scale Interacting Systems by Mean Field Method. In *Fifth International Conference on Quantitative Evaluation of Systems*, pages 215–224. IEEE, Sept. 2008. ISBN 978-0-7695-3360-5. doi: 10.1109/QEST.2008.47. URL `http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4634974`. 4, 27, 30, 140, 141

[34] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi. *Queueing networks and Markov chains.* John Wiley & Sons, 1998. 3, 18, 28

[35] L. Bortolussi. Stochastic Concurrent Constraint Programming. In *Fourth International Workshop on Quantitative Aspects of Programming Languages (QAPL)*, volume 164 of *Electronic Notes in Theoretical Computer Science*, pages 65–80, Oct. 2006. doi: 10.1016/j.entcs.2006.07.012. URL http://dx.doi.org/10.1016/j.entcs.2006.07.012. 4, 25, 140

[36] L. Bortolussi. On the Approximation of Stochastic Concurrent Constraint Programming by Master Equation. In *Sixth Workshop on Quantitative Aspects of Programming Languages (QAPL)*, volume 220 of *Electronic Notes in Theoretical Computer Science*, pages 163–180, Dec. 2008. doi: 10.1016/j.entcs.2008.11.025. URL http://dx.doi.org/10.1016/j.entcs.2008.11.025. 26

[37] L. Bortolussi. Limit Behavior of the Hybrid Approximation of Stochastic Process Algebras. In K. Al-Begain, D. Fiems, and W. J. Knottenbelt, editors, *Analytical and Stochastic Modeling Techniques and Applications (ASMTA)*, volume 6148 of *Lecture Notes in Computer Science*, pages 367–381. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-13567-5. doi: 10.1007/978-3-642-13568-2. URL http://www.springerlink.com/content/e866lj737n011l42/. 26

[38] L. Bortolussi and A. Policriti. Stochastic Concurrent Constraint Programming and Differential Equations. In *Fifth Workshop on Quantitative Aspects of Programming Languages (QAPL)*, volume 190 of *Electronic Notes in Theoretical Computer Science*, pages 27–42, Sept. 2007. doi: 10.1016/j.entcs.2007.07.003. URL http://dx.doi.org/10.1016/j.entcs.2007.07.003. 4, 25

[39] L. Bortolussi and A. Policriti. Hybrid systems and biology: continuous and discrete modeling for systems biology. In *8th International Conference on Formal Methods for Computational Systems Biology (SFM)*, pages 424–448. Springer, 2008. URL http://portal.acm.org/citation.cfm?id=1786698.1786712. 26

[40] L. Bortolussi and A. Policriti. Hybrid approximation of Stochastic Concurrent Constraint Programming. In *17th IFAC World Congress*, 2008. 26

[41] L. Bortolussi and A. Policriti. Dynamical Systems and Stochastic Programming: To Ordinary Differential Equations and Back. *Transactions on Computational Systems Biology XI*, 5750:216–267, 2009. doi: 10.1007/978-3-642-04186-0. URL http://www.springerlink.com/content/g61p177510781x65/. 25

[42] L. Bortolussi and A. Policriti. The Importance of Being (A Little Bit) Discrete. In *Second Workshop From Biology to Concurrency and Back (FBTC)*, volume 229 of *Electronic Notes in Theoretical Computer Science*, pages 75–92, Feb. 2009. doi: 10.1016/j.entcs.2009.02.006. URL http://dx.doi.org/10.1016/j.entcs.2009.02.006. 26

[43] L. Bortolussi and A. Policriti. Stochastic Programs and Hybrid Automata for (Biological) Modeling. In K. Ambos-Spies, B. Löwe, and W. Merkle, editors, *Mathematical Theory and Computational Practice*, volume 5635 of *Lecture Notes in Computer Science*, pages 37–48. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-03072-7. doi: 10.1007/978-3-642-03073-4. URL http://www.springerlink.com/content/026g745267581425/. 26

[44] L. Bortolussi and A. Policriti. Hybrid dynamics of stochastic programs. *Theoretical Computer Science*, 411(20):2052–2077, Apr. 2010. ISSN 03043975. doi: 10.1016/j.tcs.2010.02.008. URL http://dx.doi.org/10.1016/j.tcs.2010.02.008. 25, 26

[45] L. Bortolussi, V. Galpin, J. Hillston, and M. Tribastone. Hybrid Semantics for PEPA. In *Seventh International Conference on the Quantitative Evaluation of Systems*, pages 181–190. IEEE, Sept. 2010. ISBN 978-1-4244-8082-1. doi: 10.1109/QEST.2010.31. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5600389. 25

[46] H. Bowman. Analysis of a Multimedia Stream using Stochastic Process Algebra. *The Computer Journal*, 44(4):230–245, Apr. 2001. ISSN 0010-4620. doi: 10.1093/comjnl/44.4.230. URL http://comjnl.oxfordjournals.org/cgi/content/abstract/44/4/230. 18

[47] O. J. Boxma and V. Dumas. The busy period in the fluid queue. *ACM SIGMETRICS Performance Evaluation Review*, 26(1):100–110, June 1998. ISSN 01635999. doi: 10.1145/277858.277881. URL http://portal.acm.org/citation.cfm?id=277858.277881. 28

[48] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*. Society for Industrial and Applied Mathematics, 1994. 125

[49] J. T. Bradley. Semi-Markov PEPA: Modelling with Generally Distributed Actions. *International Journal of Simulation*, 6(3-4):43–51, 2005. 77, 78

[50] J. T. Bradley, N. J. Dingle, S. T. Gilmore, and W. J. Knottenbelt. Derivation of passage-time densities in PEPA models using ipc: the Imperial PEPA Compiler. In *11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems (MASCOTS)*, pages 344–351. IEEE Comput. Soc, 2003. ISBN 0-7695-2039-1. doi: 10.1109/MASCOT.2003.1240679. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1240679. 18, 57

[51] J. T. Bradley, S. T. Gilmore, and J. Hillston. Analysing distributed Internet worm attacks using continuous state-space approximation of process algebra models. *Journal of Computer and System Sciences*, 74(6):1013–1032, Sept. 2008. ISSN 00220000. doi: 10.1016/j.jcss.2007.07.005. URL http://portal.acm.org/citation.cfm?id=1399649.1399829. 24

[52] J. T. Bradley, R. A. Hayden, W. J. Knottenbelt, and T. Suto. Extracting Response Times from Fluid Analysis of Performance Models. In S. Kounev, I. Gorton, and K. Sachs, editors, *SPEC International Performance Evaluation Workshop*, volume 5119 of *Lecture Notes in Computer Science*, pages 29–43, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-69813-5. doi: 10.1007/978-3-540-69814-2. URL http://www.springerlink.com/content/d3605q665xp13017/. 9, 58

[53] J. T. Bradley, R. A. Hayden, W. J. Knottenbelt, and T. Suto. Extracting Fluid Response times from PEPA models. In *7th Workshop on Process Algebra and Stochastically Timed Activities (PASTA)*, 2008. URL http://aesop.doc.ic.ac.uk/pubs/responsetimes-fluidpepa/. 9

[54] M. Bravetti, S. T. Gilmore, C. Guidi, and M. Tribastone. Replicating web services for scalability. In *3rd Conference on Trustworthy Global Computing (TGC)*, pages 204–221. Springer, 2007. URL http://portal.acm.org/citation.cfm?id=1793591. 18

[55] M. Brin and G. Stuck. *Introduction to Dynamical Systems*. Cambridge University Press, 2002. 127

[56] D. Bruneo, M. Scarpa, A. Bobbio, D. Cerotti, and M. Gribaudo. Analytical modeling of swarm intelligence in wireless sensor networks through Markovian agents. In *Fourth International ICST Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS)*, pages 52:1–52:10. ICST, 2009. ISBN 978-963-9799-70-7. doi: 10.4108/ICST.VALUETOOLS2009.7672. URL `http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7672`. 27

[57] P. Buchholz. Hierarchical Markovian Models: Symmetries and Reduction. *Performance Evaluation*, 22(1):93–110, 1992. doi: 10.1016/0166-5316(93)E0040-C. URL `http://www.sciencedirect.com/science/article/pii/0166531693E0040C`. 16

[58] P. Buchholz. Exact and Ordinary Lumpability in Finite Markov Chains. *Journal of Applied Probability*, 31(1):59–75, 1994. URL `http://www.jstor.org/stable/3215235`. 3, 16

[59] M. L. Bujorianu and J. Lygeros. Theoretical foundations of stochastic hybrid systems. In *XVI International Symposium on Mathematical Theory of Networks and Systems, MTNS*, 2004. 26

[60] J. Campos and M. Silva. Structural Techniques and Performance Bounds of Stochastic Petri Net Models. In *Advances in Petri Nets*, Lectures Notes in Computer Science, pages 352–391, 1992. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.34.6595`. 29

[61] L. Cardelli. A Process Algebra Master Equation. In *Fourth International Conference on the Quantitative Evaluation of Systems (QEST)*, pages 219–226. IEEE, Sept. 2007. ISBN 0-7695-2883-X. doi: 10.1109/QEST.2007.26. URL `http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4338261`. 25

[62] L. Cardelli. From Processes to ODEs by Chemistry. In G. Ausiello, J. Karhumäki, G. Mauri, and L. Ong, editors, *Fifth IFIP International Conference On Theoretical Computer Science (TCS)*, volume 273 of *IFIP International Federation for Information Processing*, pages 261–281. Springer US, 2008. ISBN 978-0-387-09679-7. doi: 10.1007/978-0-387-09680-3. URL `http://www.springerlink.com/content/8g12ph2857372xm2/`. 23, 25

[63] L. Cardelli. On process rate semantics. *Theoretical Computer Science*, 391(3):190–215, Feb. 2008. ISSN 03043975. doi: 10.1016/j.tcs.2007.11.012. URL `http://dx.doi.org/10.1016/j.tcs.2007.11.012`. 25

[64] D. Cerotti. *Interacting Markovian Agents*. Ph. D., University of Torino, 2010. 27

[65] D. Cerotti, M. Gribaudo, and A. Bobbio. Presenting Dynamic Markovian Agents with a road tunnel application. In *IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 1–4. IEEE, Sept. 2009. ISBN 978-1-4244-4927-9. doi: 10.1109/MASCOT.2009.5367075. URL `http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5367075`. 27

[66] D. Cerotti, M. Gribaudo, and A. Bobbio. Disaster Propagation in Heterogeneous Media via Markovian Agents. In R. Setola and S. Geretshuber, editors, *3rd International Workshop on Critical Information Infrastructures Security (CRITIS)*, volume 5508 of *Lecture Notes in Computer Science*, pages 328–335. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-03551-7. doi: 10.1007/978-3-642-03552-4. URL http://www.springerlink.com/content/17xx274q71ug1p4r/. 27

[67] D. Cerotti, M. Gribaudo, A. Bobbio, C. Calafate, and P. Manzoni. A Markovian Agent Model for Fire Propagation in Outdoor Environments. In A. Aldini, M. Bernardo, L. Bononi, and V. Cortellessa, editors, *7th European Performance Engineering Workshop (EPEW)*, volume 6342 of *Lecture Notes in Computer Science*, pages 131–146. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-15783-7. doi: 10.1007/978-3-642-15784-4. URL http://www.springerlink.com/content/91188778014g3686/. 27

[68] A. Chaintreau, J.-Y. Le Boudec, and N. Ristanovic. The age of gossip. In *Eleventh international joint conference on measurement and modeling of computer systems (SIGMETRICS)*, pages 109–120, New York, New York, USA, June 2009. ACM Press. ISBN 9781605585116. doi: 10.1145/1555349.1555363. URL http://portal.acm.org/citation.cfm?id=1555349.1555363. 27

[69] H. Chen and D. D. Yao. *Fundamentals of queueing networks: performance, asymptotics, and optimization*. Springer, 2001. 28

[70] T. Chen, T. Han, J.-P. Katoen, and A. Mereacre. Model Checking of Continuous-Time Markov Chains Against Timed Automata Specifications. *Logical Methods in Computer Science*, 7(1):1–34, Mar. 2011. URL http://www.www.lmcs-online.org/ojs/viewarticle.php?id=644&layout=abstract. 26

[71] G. Chiola. Manual and Automatic Exploitation of Symmetries in SPN Models. In *Proceedings of the 19th International Conference on Application and Theory of Petri Nets*, pages 28–43. Springer, June 1998. ISBN 3-540-64677-9. URL http://portal.acm.org/citation.cfm?id=647745.734041. 16

[72] J.-W. Cho, J.-Y. Le Boudec, and Y. Jiang. On the validity of the fixed point equation and decoupling assumption for analyzing the 802.11 MAC protocol. *ACM SIGMETRICS Performance Evaluation Review*, 38(2):36, Oct. 2010. ISSN 01635999. doi: 10.1145/1870178.1870191. URL http://portal.acm.org/citation.cfm?id=1870178.1870191. 27

[73] J.-W. Cho, J.-Y. Le Boudec, and Y. Jiang. On the Validity of the Fixed Point Equation and Decoupling Assumption for Analyzing the 802.11 MAC Protocol. In *Twelfth Workshop on Mathematical performance Modeling and Analysis (MAMA)*, 2010. 27

[74] G. Ciardo and A. S. Miner. SMART: The Stochastic Model checking Analyzer for Reliability and Timing. In *First International Conference on Quantitative Evaluation of Systems (QEST)*, pages 338–339. IEEE Computer Society, Sept. 2004. doi: 10.1109/QEST.2004.10018. URL http://www.computer.org/portal/web/csdl/doi/10.1109/QEST.2004.10018. 17

[75] G. Ciardo, D. M. Nicol, and K. S. Trivedi. Discrete-event simulation of fluid stochastic Petri nets. *IEEE Transactions on Software Engineering*, 25(2):207–217, 1999. ISSN

00985589. doi: 10.1109/32.761446. URL http://portal.acm.org/citation.cfm?id=630823.631240. 28, 29

[76] G. Ciardo, M. Forno, P. L. E. Grieco, and A. S. Miner. Comparing implicit representations of large CTMCs. In *Numerical Solution of Markov Chains (NSMC)*, pages 323–327, 2003. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.8333. 17

[77] F. Ciocchetta and J. Hillston. Bio-PEPA: A framework for the modelling and analysis of biological systems. *Theoretical Computer Science*, 410(33-34):3065–3084, Aug. 2009. ISSN 03043975. doi: 10.1016/j.tcs.2009.02.037. URL http://dx.doi.org/10.1016/j.tcs.2009.02.037. 23

[78] A. Clark. Response-time Profiles for PEPA models compiled to ODEs. In *8th Workshop on Process Algebra and Stochastically Timed Activities (PASTA)*, pages 57–65, 2009. 58

[79] A. Clark and S. T. Gilmore. State-Aware Performance Analysis with eXtended Stochastic Probes. In N. Thomas and C. Juiz, editors, *5th European Performance Engineering Workshop (EPEW)*, volume 5261 of *Lecture Notes in Computer Science*, pages 125–140. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-87411-9. doi: 10.1007/978-3-540-87412-6. URL http://www.springerlink.com/content/tm03531636056818/. 76, 82, 83

[80] A. Clark and S. T. Gilmore. Transformations in PEPA Models and Stochastic Probe Placement. In *25th UK Performance Engineering Workshop (UKPEW)*, pages 1–16, 2009. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.154.8799. 76, 82, 91

[81] A. Clark, A. Duguid, S. T. Gilmore, and M. Tribastone. Partial Evaluation of PEPA Models for Fluid-Flow Analysis. In N. Thomas and C. Juiz, editors, *Proceedings of the 5th European Performance Engineering Workshop on Computer Performance Engineering (EPEW)*, volume 5261 of *Lecture Notes in Computer Science*, pages 2–16. Springer Berlin Heidelberg, Aug. 2008. ISBN 978-3-540-87411-9. doi: 10.1007/978-3-540-87412-6. URL http://portal.acm.org/citation.cfm?id=1433749.1433752. 24, 25, 32, 55, 65, 142

[82] A. Clark, S. T. Gilmore, and M. Tribastone. Quantitative Analysis of Web Services Using SRMC. In M. Bernardo, L. Padovani, and G. Zavattaro, editors, *Formal Methods for Web Services*, volume 5569 of *Lecture Notes in Computer Science*, pages 296–339. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-01917-3. doi: 10.1007/978-3-642-01918-0. URL http://www.springerlink.com/content/124751nlq5575328/. 24

[83] P. R. D'Argenio, H. Hermanns, J.-P. Katoen, and R. Klaren. MoDeST - A Modelling and Description Language for Stochastic Timed Systems. In *Proceedings of the Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification*, pages 87–104, Sept. 2001. ISBN 3-540-42556-X. URL http://portal.acm.org/citation.cfm?id=645776.668428. 77

[84] R. W. R. Darling. Fluid Limits of Pure Jump Markov Processes: a Practical Guide. Oct. 2002. URL http://arxiv.org/abs/math/0210109. 117

[85] R. W. R. Darling and J. R. Norris. Differential equation approximations for Markov chains. *Probability Surveys*, 5:37–79, 2008. ISSN 1549-5787. URL http://projecteuclid.org/euclid.ps/1208958281. 5, 11, 117, 118, 189

[86] R. David and H. Alla. *Discrete, Continuous, and Hybrid Petri Nets.* Springer Berlin Heidelberg, second edition, 2010. 4, 29, 30, 141

[87] M. H. A. Davis. *Markov models and optimization.* Chapman & Hall/CRC, 1993. 26

[88] T. Dayar and W. J. Stewart. Quasi Lumpability, Lower-Bounding Coupling Matrices, and Nearly Completely Decomposable Markov Chains. *SIAM Journal on Matrix Analysis and Applications*, 18(2):482, Apr. 1997. ISSN 08954798. doi: 10.1137/S0895479895294277. URL `http://portal.acm.org/citation.cfm?id=250263.250298`. 17

[89] L. de Alfaro, M. Z. Kwiatkowska, G. Norman, D. Parker, and R. Segala. Symbolic Model Checking of Probabilistic Processes Using MTBDDs and the Kronecker Representation. In S. Graf and M. Schwartzbach, editors, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 1785 of *Lecture Notes in Computer Science*, pages 395–410. Springer Berlin Heidelberg, June 2000. ISBN 978-3-540-67282-1. doi: 10.1007/3-540-46419-0. URL `http://www.springerlink.com/content/dgffnxjd43dap5p8/`. 17

[90] D. Deavours, G. Clark, T. Courtney, D. Daly, S. Derisavi, J. Doyle, W. Sanders, and P. Webster. The Mobius framework and its implementation. *IEEE Transactions on Software Engineering*, 28(10):956–969, Oct. 2002. ISSN 0098-5589. doi: 10.1109/TSE.2002.1041052. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1041052`. 17, 76

[91] J. Ding. *Structural and Fluid Analysis for Large Scale PEPA Models âĂŤ With Applications to Content Adaptation Systems.* Ph. D., University of Edinburgh, 2010. 24, 25, 31, 32, 65, 119, 121, 126, 132, 142

[92] J. Ding, J. Hillston, and D. Laurenson. Performance Modelling of Content Adaptation for a Personal Distributed Environment. *Wireless Personal Communications*, 48(1):93–112, Nov. 2007. ISSN 0929-6212. doi: 10.1007/s11277-007-9427-4. URL `http://www.springerlink.com/content/h65454305n8j4431/`. 24

[93] N. J. Dingle and W. J. Knottenbelt. Automated Customer-Centric Performance Analysis of Generalised Stochastic Petri Nets Using Tagged Tokens. In *Third International Workshop on the Practical Application of Stochastic Modelling (PASM)*, volume 232 of *Electronic Notes in Theoretical Computer Science*, pages 75–88, Amsterdam, Mar. 2009. Elsevier Science Publishers B. V. doi: 10.1016/j.entcs.2009.02.051. URL `http://portal.acm.org/citation.cfm?id=1518330.1518536`. 58

[94] N. J. Dingle, P. G. Harrison, and W. J. Knottenbelt. Uniformization and hypergraph partitioning for the distributed computation of response time densities in very large Markov models. *Journal of Parallel and Distributed Computing*, 64(8):908, 2004. ISSN 0743-7315. URL `http://portal.acm.org/citation.cfm?id=1032108`. 3, 58

[95] S. Donatelli, S. Haddad, and J. Sproston. Model Checking Timed and Stochastic Properties with CSL^{TA}. *IEEE Transactions on Software Engineering*, 35(2):224–240, Mar. 2009. ISSN 0098-5589. doi: 10.1109/TSE.2008.108. URL `http://portal.acm.org/citation.cfm?id=1525651.1525791`. 76, 82, 83, 113

[96] A. Duguid. Coping with the Parallelism of BitTorrent: Conversion of PEPA to ODEs in Dealing with State Space Explosion. In E. Asarin and P. Bouyer, editors, *Formal Modeling and Analysis of Timed Systems*, volume 4202 of *Lecture Notes in Computer Science*,

pages 156–170. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-45026-9. doi: 10.1007/11867340. URL http://www.springerlink.com/content/v502l45246r63t61/. 18, 24

[97] S. N. Ethier and T. G. Kurtz. *Markov Processes: Characterization and Convergence*. Wiley, 2005. 119, 120, 137, 186, 187

[98] A. Feldmann and W. Whitt. Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. In *Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1096–1104. IEEE Comput. Soc. Press, 1997. ISBN 0-8186-7780-5. doi: 10.1109/INFCOM.1997.631130. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=631130. 2

[99] A. J. Field and P. G. Harrison. An approximate compositional approach to the analysis of fluid queue networks. *Performance Evaluation*, 64(9-12):1137–1152, Oct. 2007. ISSN 01665316. doi: 10.1016/j.peva.2007.06.025. URL http://portal.acm.org/citation.cfm?id=1284907.1285036. 28

[100] J.-M. Fourneau, L. Kloul, and F. Valois. Performance modelling of hierarchical cellular networks using PEPA. *Performance Evaluation*, 50(2):83, 2002. ISSN 0166-5316. URL http://portal.acm.org/citation.cfm?id=638118. 18

[101] G. Franceschinis and R. R. Muntz. Bounds for quasi-lumpable Markov chains. *Performance Evaluation*, 20(1-3):223–243, May 1994. ISSN 01665316. doi: 10.1016/0166-5316(94)90015-9. URL http://portal.acm.org/citation.cfm?id=198602.198636. 17

[102] J. Freudiger, M. H. Manshaei, J.-Y. Le Boudec, and J.-P. Hubaux. On the age of pseudonyms in mobile ad hoc networks. In *29th Conference on Information Communications (INFOCOM)*, pages 1577–1585. IEEE Press, 2010. ISBN 978-1-4244-5836-3. URL http://portal.acm.org/citation.cfm?id=1833515.1833737. 27

[103] M. Fujita, P. C. McGeer, and J. C. Y. Yang. Multi-Terminal Binary Decision Diagrams: An Efficient Data Structure for Matrix Representation. In *Formal Methods in System Design*, volume 10, pages 149–169. Springer Netherlands, 1997. doi: 10.1023/A:1008647823331. URL http://www.springerlink.com/content/pn0805u9102624r2/. 3, 17

[104] N. Gast and B. Gaujal. A mean field approach for optimization in particle systems and applications. In *Fourth International ICST Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS)*, pages 39:1–39:10. ICST, 2009. ISBN 978-963-9799-70-7. doi: 10.4108/ICST.VALUETOOLS2009.7477. URL http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7477. 28

[105] N. Gast and B. Gaujal. A mean field approach for optimization in discrete time. *Discrete Event Dynamic Systems*, 21(1):63–101, Oct. 2010. ISSN 0924-6703. doi: 10.1007/s10626-010-0094-3. URL http://www.springerlink.com/content/021552ltg0646747/. 28

[106] N. Gast and B. Gaujal. Mean field limit of non-smooth systems and differential inclusions. *ACM SIGMETRICS Performance Evaluation Review*, 38(2):30, Oct. 2010. ISSN 01635999. doi: 10.1145/1870178.1870189. URL http://portal.acm.org/citation.cfm?id=1870178.1870189. 28, 152

[107] N. Gast, B. Gaujal, and J.-Y. Le Boudec. Mean field for Markov Decision Processes : from Discrete to Continuous Optimization. Technical report, 2010. URL http://arxiv.org/abs/1004.2342. 27

[108] N. Geisweiller, J. Hillston, and M. Stenico. Relating continuous and discrete PEPA models of signalling pathways. *Theoretical Computer Science*, 404(1-2):97–111, Sept. 2008. ISSN 03043975. doi: 10.1016/j.tcs.2008.04.012. URL http://dx.doi.org/10.1016/j.tcs.2008.04.012. 24, 119

[109] R. German. *Performance Analysis of Communication Systems with Non-Markovian Stochastic Petri Nets*. John Wiley & Sons, June 2000. ISBN 0471492582. URL http://portal.acm.org/citation.cfm?id=518204. 83

[110] C. S. Gillespie. Moment-closure approximations for mass-action models. *IET Systems Biology*, 3(1):52–58, Jan. 2009. ISSN 1751-8849. doi: 10.1049/iet-syb:20070031. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4762250. 148

[111] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, Dec. 1977. ISSN 0022-3654. doi: 10.1021/j100540a008. URL http://dx.doi.org/10.1021/j100540a008. 3

[112] S. T. Gilmore, J. Hillston, and M. Ribaudo. An efficient algorithm for aggregating PEPA models. *IEEE Transactions on Software Engineering*, 27(5):449–464, May 2001. ISSN 00985589. doi: 10.1109/32.922715. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=922715. 3, 16

[113] G. H. Golub and C. F. van Loan. *Matrix computations*. John Hopkins University Press, 1996. 14

[114] J. Gómez-Serrano, C. Graham, and J.-Y. Le Boudec. The Bounded Confidence Model Of Opinion Dynamics. Technical report, June 2010. URL http://arxiv.org/abs/1006.3798. 27

[115] C. Graham. *Kinetic limits for large communication networks*, pages 317–330. Modeling in applied sciences: a kinetic theory approach. Springer, 2000. 4, 27

[116] C. Graham. Chaoticity on path space for a queueing network with selection of the shortest queue among several. *Journal of Applied Probability*, 37(1):198–211, Mar. 2000. ISSN 1475-6072. URL http://projecteuclid.org/euclid.jap/1014842277. 4, 27

[117] C. Graham and S. Méléard. Stochastic particle approximations for generalized Boltzmann models and convergence estimates. *Annals of Probability*, 25(1):115–132, 1997. ISSN 0091-1798. URL http://cat.inist.fr/?aModele=afficheN&cpsidt=2715716. 4, 27, 140

[118] C. Graham and P. Robert. Interacting multi-class transmissions in large stochastic networks. *The Annals of Applied Probability*, 19(6):2334–2361, Dec. 2009. ISSN 1050-5164. URL http://projecteuclid.org/euclid.aoap/1259158774. 4, 27

[119] W. K. Grassmann. Means and variances of time averages in Markovian environments. *European Journal of Operational Research*, 31(1):132–139, July 1987. ISSN 03772217. doi: 10.1016/0377-2217(87)90147-0. URL http://dx.doi.org/10.1016/0377-2217(87)90147-0. 15

[120] W. K. Grassmann, M. I. Taksar, and D. P. Heyman. Regenerative Analysis and Steady State Distributions for Markov Chains. *Operations Research*, 33(5):1107 – 1116, 1985. URL http://www.jstor.org/stable/170856. 14

[121] M. Gribaudo. Analysis of Large Populations of Interacting Objects with Mean Field and Markovian Agents. In J. T. Bradley, editor, *6th European Performance Engineering Workshop (EPEW)*, volume 5652 of *Lecture Notes in Computer Science*, pages 218–219. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-02923-3. doi: 10.1007/978-3-642-02924-0. URL http://www.springerlink.com/content/h48275wx4797542m/. 27

[122] M. Gribaudo, D. Cerotti, and A. Bobbio. Analysis of On-off policies in Sensor Networks Using Interacting Markovian Agents. In *Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 300–305. IEEE, Mar. 2008. ISBN 978-0-7695-3113-7. doi: 10.1109/PERCOM.2008.100. URL http://www.computer.org/portal/web/csdl/doi/10.1109/PERCOM.2008.100. 27

[123] G. Hachtel, E. Macii, A. Pardo, and F. Somenzi. Probabilistic Analysis of Large Finite State Machines. In *31st Conference on Design Automation*, pages 270–275, 1994. doi: 10.1109/DAC.1994.204110. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1600383. 17

[124] M. D. Harrison and M. Massink. Modelling Interactive Experience, Function and Performance in Ubiquitous Systems. In *Fourth International Workshop on the Practical Application of Stochastic Modelling (PASM)*, volume 261 of *Electronic Notes in Theoretical Computer Science*, pages 23–42, Feb. 2010. doi: 10.1016/j.entcs.2010.01.004. URL http://dx.doi.org/10.1016/j.entcs.2010.01.004. 24

[125] P. G. Harrison. Turning back time in Markovian process algebra. *Theoretical Computer Science*, 290(3):1947–1986, Jan. 2003. ISSN 03043975. doi: 10.1016/S0304-3975(02)00375-4. URL http://dx.doi.org/10.1016/S0304-3975(02)00375-4. 3, 18

[126] P. G. Harrison. Product-forms and functional rates. *Performance Evaluation*, 66(11):660–663, Nov. 2009. ISSN 01665316. doi: 10.1016/j.peva.2009.05.003. URL http://dx.doi.org/10.1016/j.peva.2009.05.003. 3

[127] P. G. Harrison and T. T. Lee. Separable equilibrium state probabilities via time reversal in Markovian process algebra. *Theoretical Computer Science*, 346(1):161–182, Nov. 2005. ISSN 03043975. doi: 10.1016/j.tcs.2005.08.007. URL http://dx.doi.org/10.1016/j.tcs.2005.08.007. 3

[128] P. G. Harrison and B. Strulo. SPADES - a process algebra for discrete event simulation. *Journal of Logic and Computation*, 10(1):3–42, Feb. 2000. ISSN 0955-792X. doi: 10.1093/logcom/10.1.3. URL http://logcom.oxfordjournals.org/cgi/content/abstract/10/1/3. 77

[129] R. A. Hayden. *Addressing the state space explosion problem for PEPA models through fluid-flow approximation.* Masters Thesis, Imperial College London, 2007. URL http://pubs.doc.ic.ac.uk/fluid-spa-modelling/fluid-spa-modelling.pdf. 26, 32

[130] R. A. Hayden. Convergence of ODE approximations and bounds on performance models in the steady-state. In *9th Workshop on Process Algebra and Stochastically Timed Activities (PASTA)*, pages 49–56, 2010. URL `http://aesop.doc.ic.ac.uk/pubs/ode-stationary-bounds/`. 10

[131] R. A. Hayden. Ph. D. thesis additional material, 2011. URL `http://rhayden.me.uk/thesis`. 6, 38, 40, 52, 60, 64, 70, 109, 144

[132] R. A. Hayden. Mean-field approximations for performance models with generally-timed transitions. *To appear in ACM SIGMETRICS Performance Evaluation Review*, 2011. 7, 149

[133] R. A. Hayden. Toward efficient parameter sweeping and optimisation of fluid performance models by Taylor-model based symbolic numerical integration. In *10th Workshop on Process Algebra and Stochastically Timed Activities (PASTA)*, pages 19–26, 2011. URL `http://aesop.doc.ic.ac.uk/pubs/ode-sweep-taylor-models/`. 149

[134] R. A. Hayden and J. T. Bradley. Fluid-flow solutions in PEPA to the state space explosion problem. In *6th Workshop on Process Algebra and Stochastically Timed Activities (PASTA)*, pages 25–37, 2007. URL `http://aesop.doc.ic.ac.uk/pubs/fluid-spa-synopsis/`. 26

[135] R. A. Hayden and J. T. Bradley. Fluid semantics for passive stochastic process algebra cooperation. In *3rd International ICST Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS)*. ICST, Oct. 2008. URL `http://eudl.eu/article.php?id=4329`. 9

[136] R. A. Hayden and J. T. Bradley. ODE-based general moment approximations for PEPA. In *7th Workshop on Process Algebra and Stochastically Timed Activities (PASTA)*, 2008. URL `http://aesop.doc.ic.ac.uk/pubs/ode-moments-pepa/`. 9, 26

[137] R. A. Hayden and J. T. Bradley. Shared Transaction Markov Chains for fluid analysis of massively parallel systems. In *IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 1–12. IEEE, Sept. 2009. ISBN 978-1-4244-4927-9. doi: 10.1109/MASCOT.2009.5367050. URL `http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5367050`. 8, 151, 152

[138] R. A. Hayden and J. T. Bradley. A functional central limit theorem for PEPA. In *8th Workshop on Process Algebra and Stochastically Timed Activities*, pages 13–23, 2009. URL `http://aesop.doc.ic.ac.uk/pubs/pepa-functional-central-limit/`. 9, 148

[139] R. A. Hayden and J. T. Bradley. A fluid analysis framework for a Markovian process algebra. *Theoretical Computer Science*, 411(22-24):2260–2297, May 2010. ISSN 03043975. doi: 10.1016/j.tcs.2010.02.001. URL `http://dx.doi.org/10.1016/j.tcs.2010.02.001`. 4, 7, 24, 31, 33, 34, 47, 141

[140] R. A. Hayden and J. T. Bradley. Evaluating fluid semantics for passive stochastic process algebra cooperation. *Performance Evaluation*, 67(4):260–284, Apr. 2010. ISSN 01665316. doi: 10.1016/j.peva.2009.08.010. URL `http://portal.acm.org/citation.cfm?id=1750584.1750674`. 7, 9, 24, 33, 119

[141] R. A. Hayden, J. T. Bradley, and A. Clark. Performance Specification and Evaluation with Unified Stochastic Probes and Fluid Analysis. *Submitted to Transactions on Software Engineering*, 2010. 7, 55

[142] R. A. Hayden, A. Stefanek, and J. T. Bradley. Fluid computation of passage time distributions in large Markov models. *To appear in Theoretical Computer Science*, 2011. doi: 10.1016/j.tcs.2011.07.017. URL `http://aesop.doc.ic.ac.uk/pubs/fluid-passage-time/`. 6, 7, 9, 23, 55, 65, 126, 143, 149

[143] H. Hermanns. *Interactive Markov Chains*, volume 2428 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, Sept. 2002. ISBN 978-3-540-44261-5. doi: 10.1007/3-540-45804-2. URL `http://www.springerlink.com/content/978-3-540-44261-5/`. 151

[144] H. Hermanns, M. Rettelbach, and T. Weiss. Formal Characterisation of Immediate Actions in SPA with Nondeterministic Branching. *The Computer Journal*, 38(7): 530–541, July 1995. ISSN 0010-4620. doi: 10.1093/comjnl/38.7.530. URL `http://comjnl.oxfordjournals.org/cgi/content/abstract/38/7/530`. 77

[145] H. Hermanns, M. Z. Kwiatkowska, G. Norman, D. Parker, and M. Siegle. On the use of MTBDDs for performability analysis and verification of stochastic systems. *Journal of Logic and Algebraic Programming*, 56(1-2):23–67, Aug. 2003. ISSN 15678326. doi: 10.1016/S1567-8326(02)00066-8. URL `http://dx.doi.org/10.1016/S1567-8326(02)00066-8`. 3, 17

[146] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, Cambridge, 1996. ISBN 9780511569951. doi: 10.1017/CBO9780511569951. URL `http://ebooks.cambridge.org/ref/id/CBO9780511569951`. 4, 18, 30, 140

[147] J. Hillston. Fluid flow approximation of PEPA models. In *Second International Conference on the Quantitative Evaluation of Systems (QEST)*, pages 33–42. IEEE, Sept. 2005. ISBN 0-7695-2427-3. doi: 10.1109/QEST.2005.12. URL `http://www.computer.org/portal/web/csdl/doi/10.1109/QEST.2005.12`. 4, 7, 23, 24, 25, 31, 32, 56, 121, 140

[148] D. R. W. Holton. A PEPA Specification of an Industrial Production Cell. *The Computer Journal*, 38(7):542–551, July 1995. ISSN 0010-4620. doi: 10.1093/comjnl/38.7.542. URL `http://comjnl.oxfordjournals.org/cgi/content/abstract/38/7/542`. 18

[149] G. Horton, V. G. Kulkarni, D. M. Nico, and K. S. Trivedi. Fluid stochastic Petri nets: Theory, applications, and solution techniques. *European Journal of Operational Research*, 105(1):184–201, Feb. 1998. ISSN 03772217. doi: 10.1016/S0377-2217(97)00028-3. URL `http://dx.doi.org/10.1016/S0377-2217(97)00028-3`. 28

[150] S. Ioannidis. *Scalable and Reliable Searching in Unstructured Peer-to-Peer Systems*. Ph. D., University of Toronto, 2009. 27

[151] S. Ioannidis and P. Marbach. Absence of Evidence as Evidence of Absence: A Simple Mechanism for Scalable P2P Search. In *28th IEEE Conference on Computer Communications (INFOCOM)*, pages 576–584. IEEE, Apr. 2009. ISBN 978-1-4244-3512-8. doi: 10.1109/INFCOM.2009.5061964. URL `http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5061964`. 27

[152] S. Ioannidis and L. Massoulié. Surfing the Blogosphere: Optimal Personalized Strategies for Searching the Web. In *29th IEEE Conference on Computer Communications (INFOCOM)*, pages 1–9. IEEE, Mar. 2010. ISBN 978-1-4244-5836-3. doi: 10.1109/INFCOM.2010.5462079. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5462079. 27

[153] S. Ioannidis, L. Massoulié, and A. Chaintreau. Distributed caching over heterogeneous mobile networks. *ACM SIGMETRICS Performance Evaluation Review*, 38(1):311–322, June 2010. ISSN 01635999. doi: 10.1145/1811099.1811075. URL http://portal.acm.org/citation.cfm?id=1811099.1811075. 27

[154] J. R. Jackson. Networks of Waiting Lines. *Operations Research*, 5(4):518–521, Aug. 1957. ISSN 0030-364X. doi: 10.1287/opre.5.4.518. URL http://or.journal.informs.org/cgi/content/abstract/5/4/518. 18

[155] M. Johansson and A. Rantzer. Computation of piecewise quadratic Lyapunov functions for hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):555–559, Apr. 1998. ISSN 00189286. doi: 10.1109/9.664157. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=664157. 126

[156] J. Júlvez, L. Recalde, and M. Silva. Steady-state performance evaluation of continuous mono-T-semiflow Petri nets. *Automatica*, 41(4):605–616, Apr. 2005. ISSN 00051098. doi: 10.1016/j.automatica.2004.11.007. URL http://dx.doi.org/10.1016/j.automatica.2004.11.007. 29

[157] O. Kallenberg. *Foundations of Modern Probability*. Springer, 2002. 11, 129

[158] J.-P. Katoen, M. Z. Kwiatkowska, G. Norman, and D. Parker. Faster and Symbolic CTMC Model Checking. In L. Alfaro and S. Gilmore, editors, *Process Algebra and Probabilistic Methods. Performance Modelling and Verification*, volume 2165 of *Lecture Notes in Computer Science*, pages 23–38. Springer Berlin Heidelberg, Aug. 2001. ISBN 978-3-540-42556-4. doi: 10.1007/3-540-44804-7. URL http://www.springerlink.com/content/8gc965bx79hdxr3b/. 17

[159] J.-P. Katoen, D. Klink, M. Leucker, and V. Wolf. Three-valued abstraction for continuous-time Markov chains. In *19th International Conference on Computer Aided Verification (CAV)*, pages 311–324. Springer, July 2007. ISBN 978-3-540-73367-6. URL http://portal.acm.org/citation.cfm?id=1770351.1770401. 17

[160] J. Keilson. *Markov chain models: rarity and exponentiality*. Springer Verlag, 1979. 15

[161] J. G. Kemeny and J. L. Snell. *Finite Markov chains*. Springer Verlag, 1976. 3, 16

[162] G. Kesidis, T. Konstantopoulos, and P. Sousi. A stochastic epidemiological model and a deterministic limit for BitTorrent-like peer-to-peer file-sharing networks. In *2nd Workshop on Network Control and Optimization (NET-COOP)*, pages 26–36, Nov. 2008. URL http://arxiv.org/abs/0811.1003. 25

[163] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, third edition, 2002. 124

[164] W. J. Knottenbelt and P. G. Harrison. Distributed Disk-based Solution Techniques for Large Markov Models. In *3rd Intl. Conference on the Numerical Solution of Markov Chains (NSMC)*, pages 58–75, 1999. 76, 113

[165] M. Kuntz and M. Siegle. Deriving Symbolic Representations from Stochastic Process Algebras. In *Second Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification*, pages 188–206. Springer, July 2002. ISBN 3-540-43913-7. URL http://portal.acm.org/citation.cfm?id=645777.668443. 3, 17

[166] T. G. Kurtz. Solutions of Ordinary Differential Equations as Limits of Pure Jump Markov Processes. *Journal of Applied Probability*, 7(1):49–58, 1970. URL http://www.jstor.org/stable/3212147. 24, 25, 27, 119

[167] T. G. Kurtz. Limit Theorems for Sequences of Jump Markov Processes Approximating Ordinary Differential Processes. *Journal of Applied Probability*, 8(2):344–356, 1971. URL http://www.jstor.org/stable/3211904. 119

[168] T. G. Kurtz. The Relationship between Stochastic and Deterministic Models for Chemical Reactions. *The Journal of Chemical Physics*, 57(7):2976, Sept. 1972. ISSN 00219606. doi: 10.1063/1.1678692. URL http://link.aip.org/link/doi/10.1063/1.1678692/html. 4

[169] T. G. Kurtz. Strong approximation theorems for density dependent Markov chains. *Stochastic Processes and their Applications*, 6(3):223–240, Feb. 1978. ISSN 03044149. doi: 10.1016/0304-4149(78)90020-0. URL http://dx.doi.org/10.1016/0304-4149(78)90020-0. 119

[170] H. J. Kushner. *Heavy traffic analysis of controlled queueing and communications networks*. Springer, 2001. 28

[171] M. Z. Kwiatkowska, G. Norman, and D. Parker. PRISM: Probabilistic Symbolic Model Checker. In T. Field, P. G. Harrison, J. Bradley, and U. Harder, editors, *Computer Performance Evaluation: Modelling Techniques and Tools*, volume 2324 of *Lecture Notes in Computer Science*, pages 113–140. Springer Berlin Heidelberg, Apr. 2002. ISBN 978-3-540-43539-6. doi: 10.1007/3-540-46029-2. URL http://www.springerlink.com/content/1qmt7m2uw2ullpvt/. 17, 76

[172] M. Z. Kwiatkowska, G. Norman, and D. Parker. Probabilistic symbolic model checking with PRISM: a hybrid approach. *International Journal on Software Tools for Technology Transfer*, 6(2):128–142, Apr. 2004. ISSN 1433-2779. doi: 10.1007/s10009-004-0140-2. URL http://portal.acm.org/citation.cfm?id=1014989.1014992. 3, 17, 140

[173] M. Z. Kwiatkowska, G. Norman, and D. Parker. PRISM: probabilistic model checking for performance and reliability analysis. *ACM SIGMETRICS Performance Evaluation Review*, 36(4):40, Mar. 2009. ISSN 01635999. doi: 10.1145/1530873.1530882. URL http://portal.acm.org/citation.cfm?id=1530873.1530882. 3, 113

[174] M. Z. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of Probabilistic Real-time Systems. In *23rd International Conference on Computer Aided Verification (CAV)*, volume 6806 of *Lecture Notes in Computer Science*, pages 585–591. Springer, 2011. doi: 10.1007/978-3-642-22110-1_47. URL http://www.springerlink.com/content/94061gt875727x42/. 3

[175] K. Lampka, M. Siegle, J. Ossowski, and C. Baier. Partially-shared zero-suppressed multi-terminal BDDs: concept, algorithms and applications. *Formal Methods in System Design*, 36(3):198–222, June 2010. ISSN 0925-9856. doi: 10.1007/s10703-010-0095-8. URL http://portal.acm.org/citation.cfm?id=1861250.1861268. 3, 17

[176] S. S. Lavenberg and M. Reiser. Stationary State Probabilities at Arrival Instants for Closed Queueing Networks with Multiple Types of Customers. *Journal of Applied Probability*, 17(4):1048 – 1061, 1980. URL http://www.jstor.org/stable/3213214. 18

[177] G. F. Lawler. *Introduction to Stochastic Processes*. Chapman and Hall, second edition, 2006. 2, 11

[178] J.-Y. Le Boudec. *Performance Evaluation of Computer and Communication Systems*. EPFL Press, Lausanne, Switzerland, 2010. 101

[179] J.-Y. Le Boudec, D. McDonald, and J. Mundinger. A Generic Mean Field Convergence Result for Systems of Interacting Objects. In *Fourth International Conference on the Quantitative Evaluation of Systems (QEST)*, pages 3–18. IEEE, Sept. 2007. ISBN 0-7695-2883-X. doi: 10.1109/QEST.2007.8. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4338232. 4, 27, 28, 140

[180] C. H. Lee, K.-H. Kim, and P. Kim. A moment closure method for stochastic reaction networks. *The Journal of Chemical Physics*, 130(13):134107–15, Apr. 2009. ISSN 1089-7690. doi: 10.1063/1.3103264. URL http://link.aip.org/link/doi/10.1063/1.3103264/html. 148

[181] J. Little. A Proof for the Queuing Formula: L= λ W. *Operations Research*, 9(3):383–387, 1961. ISSN 0030364X. doi: 10.2307/167570. URL http://www.citeulike.org/user/benta/article/4544563. 25, 65, 142

[182] M. Massink, D. Latella, A. Bracciali, and M. D. Harrison. A Scalable Fluid Flow Process Algebraic Approach to Emergency Egress Analysis. In *8th IEEE International Conference on Software Engineering and Formal Methods*, pages 169–180. IEEE, Sept. 2010. ISBN 978-1-4244-8289-4. doi: 10.1109/SEFM.2010.22. URL http://www.computer.org/portal/web/csdl/doi/10.1109/SEFM.2010.22. 24

[183] M. Massink, D. Latella, A. Bracciali, and J. Hillston. A Combined Process Algebraic, Agent and Fluid Flow Approach to Emergent Crowd Behaviour. Technical report, Italian National Research Council, 2010. URL http://fmt.isti.cnr.it/WEBPAPER/2010-TR-025.pdf. 31, 141

[184] M. Massink, D. Latella, A. Bracciali, and J. Hillston. Modelling Non-linear Crowd Dynamics in Bio-PEPA. In D. Giannakopoulou and F. Orejas, editors, *Fundamental Approaches to Software Engineering*, volume 6603 of *Lecture Notes in Computer Science*, pages 96–110. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-19810-6. doi: 10.1007/978-3-642-19811-3. URL http://www.springerlink.com/content/lw5812w5k78744m8/. 31, 141

[185] L. Massoulié and M. Vojnovic. Coupon Replication Systems. *IEEE/ACM Transactions on Networking*, 16(3):603–616, June 2008. doi: 10.1109/TNET.2007.903992. URL http://portal.acm.org/citation.cfm?id=1399562.1399571. 27, 65

[186] S. Méléard. Asymptotic behaviour of some interacting particle systems; McKean-Vlasov and Boltzmann models. *Probabilistic Models for Nonlinear Partial Differential Equations*, 1627:42–95, 1996. doi: 10.1007/BFb0093175. URL `http://www.springerlink.com/content/g772652p33h7105r/`. 4, 27, 140

[187] R. K. Miller and A. N. Michel. *Ordinary differential equations.* Academic Press, 1982. 40, 124, 130

[188] R. Milner. *Communicating and Mobile Systems.* Cambridge University Press, 1999. 151

[189] C. Moler and C. F. van Loan. Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later. *SIAM Review*, 45(1):3 – 49, 2003. URL `http://www.jstor.org/stable/25054364`. 14

[190] D. Mollison. *Epidemic models: their structure and relation to data.* Cambridge University Press, 1995. 4

[191] P. A. P. Moran. A Theory of Dams with Continuous Input and a General Release Rule. *Journal of Applied Probability*, 6(1):88–98, 1969. URL `http://www.jstor.org/stable/3212278`. 29

[192] J. Mundinger and J.-Y. Le Boudec. Analysis of a reputation system for Mobile Ad-Hoc Networks with liars. *Performance Evaluation*, 65(3-4):212–226, Mar. 2008. ISSN 01665316. doi: 10.1016/j.peva.2007.05.004. URL `http://dx.doi.org/10.1016/j.peva.2007.05.004`. 27

[193] J. K. Muppala and K. S. Trivedi. Numerical transient solution of finite Markovian queueing systems. *Queueing and Related Models*, (9):262–284, 1992. 15

[194] H. R. Neave. *Statistics Tables for Mathematicians, Engineers, Economists and the Behavioural and Management Sciences.* George Allen & Unwin, 1978. 109

[195] M. Neher, K. R. Jackson, and N. S. Nedialkov. On Taylor model based integration of ODEs. *SIAM Journal on Numerical Analysis*, 45(1):236–262, 2007. 149

[196] M. F. Neuts. *Matrix-geometric solutions in stochastic models: an algorithmic approach.* Courier Dover Publications, 1981. 2

[197] T. Osogami and M. Harchol-Balter. Closed form solutions for mapping general distributions to quasi-minimal PH distributions. *Performance Evaluation*, 63(6):524–552, June 2006. ISSN 01665316. doi: 10.1016/j.peva.2005.06.002. URL `http://dx.doi.org/10.1016/j.peva.2005.06.002`. 2

[198] I. Papamichail and C. S. Adjiman. A Rigorous Global Optimization Algorithm for Problems with Ordinary Differential Equations. *Journal of Global Optimization*, 24(1):1, 2002. ISSN 0925-5001. URL `http://portal.acm.org/citation.cfm?id=596464`. 149

[199] D. Parker. *Implementation of symbolic model checking for probabilistic systems.* Ph. D., University of Birmingham, 2002. 3, 17

[200] A. Pavlov, N. van De Wouw, and H. Nijmeijer. Convergent piecewise affine systems: analysis and design Part I: continuous case. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 5391–5396. IEEE, 2005. ISBN 0-7803-9567-0.

doi: 10.1109/CDC.2005.1583019. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1583019. 125

[201] P. Piazzolla and M. Gribaudo. Analysis of Television and Cinema Productions using Mean Field Models. In *Fourth International Workshop on the Practical Application of Stochastic Modelling (PASM)*, volume 261 of *Electronic Notes in Theoretical Computer Science*, pages 91–108, Feb. 2010. doi: 10.1016/j.entcs.2010.01.007. URL http://dx.doi.org/10.1016/j.entcs.2010.01.007. 27, 30, 141

[202] B. Plateau. *De l'Evaluation du Parallélisme et de la Synchronisation*. Ph. D., Université de Paris-Sud, 1984. 17

[203] B. Plateau. On the stochastic structure of parallelism and synchronization models for distributed algorithms. *ACM SIGMETRICS Performance Evaluation Review*, 13(2):147–154, Aug. 1985. ISSN 01635999. doi: 10.1145/317786.317819. URL http://portal.acm.org/citation.cfm?id=317786.317819. 17

[204] C. Priami. Stochastic π-Calculus. *The Computer Journal*, 38(7):578–589, July 1995. ISSN 0010-4620. doi: 10.1093/comjnl/38.7.578. URL http://comjnl.oxfordjournals.org/cgi/content/abstract/38/7/578. 25

[205] D. Qiu and W. Sang. Global stability of Peer-to-Peer file sharing systems. *Computer Communications*, 31(2):212–219, Feb. 2008. doi: 10.1016/j.comcom.2007.08.012. URL http://dx.doi.org/10.1016/j.comcom.2007.08.012. 5

[206] L. Rabehasaina. Moments of a Markov-Modulated, Irreducible Network of Fluid Queues. *Journal of Applied Probability*, 43(2):510–522, 2006. URL http://www.jstor.org/stable/27595747. 28

[207] A. Reibman and K. S. Trivedi. Numerical transient analysis of Markov models. *Computers & Operations Research*, 15(1):19–36, Jan. 1988. ISSN 03050548. doi: 10.1016/0305-0548(88)90026-3. URL http://portal.acm.org/citation.cfm?id=46971.46974. 15

[208] M. Reiser and S. S. Lavenberg. Mean-Value Analysis of Closed Multichain Queuing Networks. *Journal of the ACM*, 27(2):313–322, Apr. 1980. ISSN 00045411. doi: 10.1145/322186.322195. URL http://portal.acm.org/citation.cfm?id=322186.322195. 18

[209] M. Rettelbach. Probabilistic Branching in Markovian Process Algebras. *The Computer Journal*, 38(7):590–599, July 1995. ISSN 0010-4620. doi: 10.1093/comjnl/38.7.590. URL http://comjnl.oxfordjournals.org/cgi/content/abstract/38/7/590. 77

[210] L. C. G. Rogers and D. Williams. *Diffusions, Markov Processes and Martingales, Volume 1: Foundations*. Cambridge University Press, 2000. 11, 138, 188

[211] K. R. Rohloff and T. Basar. Stochastic behavior of random constant scanning worms. In *14th International Conference on Computer Communications and Networks (ICCCN)*, pages 339–344. IEEE, 2005. doi: 10.1109/ICCCN.2005.1523881. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1523881. 27

[212] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, second edition, 2003. 14, 15

[213] A. M. Sahlodin and B. Chachuat. Convex/concave relaxations of parametric ODEs using Taylor models. *Computers & Chemical Engineering*, 35(5):857–844, Jan. 2011. ISSN 00981354. doi: 10.1016/j.compchemeng.2011.01.031. URL http://dx.doi.org/10.1016/j.compchemeng.2011.01.031. 149

[214] V. A. Saraswat. *Concurrent constraint programming*. MIT press, 1993. 25

[215] R. Schlicht and G. Winkler. A delay stochastic process with applications in molecular biology. *Journal of Mathematical Biology*, 57(5):613–48, Nov. 2008. ISSN 0303-6812. doi: 10.1007/s00285-008-0178-y. URL http://www.springerlink.com/content/4251116028104471/. 149

[216] P. J. Schweitzer. Aggregation Methods for Large Markov Chains. In *International Workshop on Computer Performance and Reliability*, pages 275–286, Sept. 1983. ISBN 0-444-86892-5. URL http://portal.acm.org/citation.cfm?id=645703.663648. 16

[217] J. K. Scott and P. I. Barton. Tight, efficient bounds on the solutions of chemical kinetics models. *Computers & Chemical Engineering*, 34(5):717–731, May 2010. ISSN 00981354. doi: 10.1016/j.compchemeng.2009.11.021. URL http://yoric.mit.edu/node/406. 121

[218] R. Serfozo. *Basics of Applied Stochastic Processes*. Springer Berlin Heidelberg, 2009. 101, 103, 138

[219] K. C. Sevcik and I. Mitrani. The Distribution of Queuing Network States at Input and Output Instants. *Journal of the ACM*, 28(2):358–371, Apr. 1981. ISSN 00045411. doi: 10.1145/322248.322257. URL http://portal.acm.org/citation.cfm?id=322248.322257. 18

[220] S. Shakkottai and R. Johari. Demand-Aware Content Distribution on the Internet. *IEEE/ACM Transactions on Networking*, 18(2):476–489, Apr. 2010. doi: 10.1109/TNET.2009.2035047. URL http://portal.acm.org/citation.cfm?id=1816262.1816273. 27, 65

[221] A. Shwartz and A. Weiss. *Large deviations for performance analysis*. Chapman and Hall, 1995. 64, 118

[222] M. Siegle. Using Structured Modelling for Efficient Performance Prediction of Parallel Systems. In *Parallel Computing: Trends and Applications – Proc. of the Int. Conf. on Advances in Parallel Computing*, pages 453–460. Elsevier, 1994. 3, 16

[223] M. Silva and L. Recalde. Petri nets and integrality relaxations: a view of continuous Petri net models. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 32(4):314–327, Nov. 2002. ISSN 1094-6977. doi: 10.1109/TSMCC.2002.806063. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1176881. 29

[224] M. Silva and L. Recalde. On fluidification of Petri Nets: from discrete to hybrid and continuous models. *Annual Reviews in Control*, 28(2):253–266, 2004. ISSN 13675788. doi: 10.1016/j.arcontrol.2004.05.002. URL http://dx.doi.org/10.1016/j.arcontrol.2004.05.002. 4, 29, 30, 141

[225] M. Silva, E. Terue, and J. Colom. Linear algebraic and linear programming techniques for the analysis of place/transition net systems. In *Lectures on Petri Nets I: Basic Models*, volume 1491 of *Lecture Notes in Computer Science*, pages 309–373. Springer Berlin / Heidelberg, 1998. doi: 10.1007/3-540-65306-6_19. URL `http://www.springerlink.com/content/y1855200p44g4hx7/`. 122

[226] A. B. Singer and P. I. Barton. Global Optimization with Nonlinear Ordinary Differential Equations. *Journal of Global Optimization*, 34(2):159, 2006. ISSN 0925-5001. URL `http://portal.acm.org/citation.cfm?id=1117154`. 149

[227] M. Sipser. *Introduction to the Theory of Computation*. PWS, Boston, 1997. 87, 88

[228] M. J. A. Smith. *Stochastic Abstraction of Programs: Towards Performance-Driven Development*. Ph. D., University of Edinburgh, 2010. URL `http://lanther.co.uk/compsci/phd/thesis.pdf`. 17

[229] A. Stefanek, R. A. Hayden, and J. T. Bradley. A new tool for the performance analysis of massively parallel computer systems. In *Eighth Workshop on Quantitative Aspects of Programming Languages (QAPL)*, volume 28 of *Electronic Proceedings in Theoretical Computer Science*, pages 159–181, 2010. doi: 10.4204/EPTCS.28.11. 6, 8, 52, 55, 56, 102, 104, 106, 143, 148

[230] A. Stefanek, R. A. Hayden, and J. T. Bradley. Hybrid analysis of large scale PEPA models. In *9th Workshop on Process Algebra and Stochastically Timed Activities (PASTA)*, pages 29–36, 2010. URL `http://aesop.doc.ic.ac.uk/pubs/hybrid-anl-pepa/`. 10

[231] A. Stefanek, R. A. Hayden, and J. T. Bradley. GPA - Tool for rapid analysis of very large scale PEPA models. In *26th UK Performance Engineering Workshop*, pages 91–101, 2010. URL `http://aesop.doc.ic.ac.uk/pubs/gpa-massive-pepa-analysis/`. 10

[232] A. Stefanek, R. A. Hayden, and J. T. Bradley. Fluid analysis of energy consumption using rewards in massively parallel Markov models. In *2nd ACM/SPEC International Conference on Performance Engineering (ICPE)*, pages 121–132, 2011. doi: 10.1145/1958746.1958767. 6, 8, 24, 143, 149

[233] A. Stefanek, R. A. Hayden, and J. T. Bradley. Capturing the energy-performance trade-off in large scale cloud computing environments. In *3rd ACM/SPEC International Conference on Performance Engineering (ICPE) (submitted)*, 2011. 9, 56, 143, 149, 151

[234] A. Stefanek, R. A. Hayden, and J. T. Bradley. Fluid computation of the performance-energy trade-off in large scale Markov models. *To appear in SIGMETRICS Performance Evaluation Review*, 2011. URL `http://aesop.doc.ic.ac.uk/pubs/fluid-performance-energy`. 6

[235] A. Stefanek, R. A. Hayden, and J. T. Bradley. GPA - A tool for fluid scalability analysis of massively parallel systems. In *8th International Conference on Quantitative Evaluation of Systems (QEST)*, pages 147–148. IEEE, 2011. 6, 8, 55, 56, 102, 104, 106

[236] W. J. Stewart. *Introduction to the numerical solution of Markov chains*. Princeton University Press, 1994. 14

[237] R. Stone and H. Xin. Supercomputer leaves competition–and users–in the dust. *Science*, 330(6005):746–7, Nov. 2010. ISSN 1095-9203. doi: 10.1126/science.330.6005.746-a. URL http://www.sciencemag.org/content/330/6005/746.1.short. 15

[238] D. Stoyan. *Comparison methods for queues and other stochastic models*. Wiley, 1983. 17

[239] D. J. Sumpter, G. B. Blanchard, and D. S. Broomhead. Ants and agents: a process algebra approach to modelling ant colony behaviour. *Bulletin of Mathematical Biology*, 63(5):951–980, Sept. 2001. ISSN 0092-8240. doi: 10.1006/bulm.2001.0252. URL http://dx.doi.org/10.1006/bulm.2001.0252. 23

[240] A.-S. Sznitman. Topics in propagation of chaos. In P.-L. Hennequin, editor, *Ecole d'Eté de Probabilités de Saint-Flour XIX1 — 1989*, volume 1464 of *Lecture Notes in Mathematics*, pages 165–251. Springer Berlin Heidelberg, 1991. ISBN 978-3-540-53841-7. doi: 10.1007/BFb0085166. URL http://www.springerlink.com/content/01r5360642v81u81/. 27

[241] A. Tari, M. Telek, and P. Buchholz. A Unified Approach to the Moments Based Distribution Estimation – Unbounded Support. In M. Bravetti, L. Kloul, and G. Zavattaro, editors, *European Performance Evaluation Workshop (EPEW)*, volume 3670 of *Lecture Notes in Computer Science*, pages 79–93. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-28701-8. doi: 10.1007/11549970. URL http://www.springerlink.com/content/pv432827164p32q0/. 65

[242] H. Tembine, J.-Y. Le Boudec, R. El-Azouzi, and E. Altman. Mean field asymptotics of Markov Decision Evolutionary Games and Teams. In *International Conference on Game Theory for Networks (GameNets)*, pages 140–150. IEEE, May 2009. ISBN 978-1-4244-4176-1. doi: 10.1109/GAMENETS.2009.5137395. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5137395. 27

[243] The MathWorks. Introduction to Linear Matrix Inequalities (Robust Control Toolbox), March 2011. URL http://www.mathworks.com/access/helpdesk/help/toolbox/robust/ug/f8-2138.html. 126

[244] N. Thomas. Using ODEs from PEPA models to derive asymptotic solutions for a class of closed queueing networks. In *8th Workshop on Process Algebra and Stochastically Timed Activities (PASTA)*, pages 49–55, Nov. 2009. 29

[245] N. Thomas and Y. Zhao. Mean Value Analysis for a Class of PEPA Models. In J. T. Bradley, editor, *European Performance Evaluation Workshop (EPEW)*, volume 5652 of *Lecture Notes in Computer Science*, pages 59–72. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-02923-3. doi: 10.1007/978-3-642-02924-0. URL http://www.springerlink.com/content/k473q13037667j21/. 18

[246] N. Thomas, J. T. Bradley, and W. J. Knottenbelt. Stochastic analysis of scheduling strategies in a Grid-based resource model. *IEE Proceedings - Software*, 151(5):232, 2004. ISSN 14625970. doi: 10.1049/ip-sen:20041091. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1368429. 18

[247] C. Tofts. A synchronous calculus of relative frequency. In J. C. M. Baeten and J. W. Klop, editors, *International Conference on Concurrency Theory (CONCUR)*, volume 458

of *Lecture Notes in Computer Science*, pages 467–480. Springer-Verlag, 1990. ISBN 3-540-53048-7. doi: 10.1007/BFb0039045. URL http://www.springerlink.com/content/x732l52q57t3616v/. 23

[248] D.-C. Tomozei and L. Massoulié. Flow Control for Cost-Efficient Peer-to-Peer Streaming. In *IEEE INFOCOM*, pages 1–9. IEEE, Mar. 2010. doi: 10.1109/INFCOM.2010.5461966. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5461966. 27

[249] M. Tribastone. *Scalable Analysis of Stochastic Process Algebra Models*. Ph. D., University of Edinburgh, 2010. 23, 24, 25, 65, 119, 142

[250] M. Tribastone, S. T. Gilmore, and J. Hillston. Scalable Differential Analysis of Process Algebra Models. *IEEE Transactions on Software Engineering*, 99(PrePrints), 2010. ISSN 0098-5589. doi: 10.1109/TSE.2010.82. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5567115. 24, 31, 32, 119

[251] K. S. Trivedi. *Probability and statistics with reliability, queueing and computer science applications*. Prentice-Hall, 1982. 13

[252] K. S. Trivedi and V. G. Kulkarni. FSPNs: Fluid stochastic Petri nets. In *Application and Theory of Petri Nets*, volume 691 of *Lectures Notes in Computer Science*, pages 24–31. Springer Berlin / Heidelberg, 1993. doi: 10.1007/3-540-56863-8_38. URL http://www.springerlink.com/content/fq301012635570p8/. 28

[253] B. Tuffin, D. S. Chen, and K. S. Trivedi. Comparison of Hybrid Systems and Fluid Stochastic Petri Nets. *Discrete Event Dynamic Systems*, 11(1):77–95, 2001. ISSN 0924-6703. doi: 10.1023/A:1008387132533. URL http://www.springerlink.com/content/px11jgm420587853/. 29

[254] A. G. Turner. Convergence of Markov processes near saddle fixed points. *The Annals of Probability*, 35(3):1141–1171, May 2007. ISSN 0091-1798. URL http://projecteuclid.org/euclid.aop/1178804325. 148

[255] W. Whitt. *Stochastic-Process Limits: An Introduction to Stochastic-Process Limits and Their Application to Queues*. Springer, 2002. 28

[256] S. Wiggins. *Introduction to applied nonlinear dynamical systems and chaos*. Springer, 2000. 127

[257] Y. Zhao and N. Thomas. Comparing Methods for the Efficient Analysis of PEPA Models of Non-repudiation Protocols. In *15th International Conference on Parallel and Distributed Systems*, pages 821–827. IEEE, 2009. ISBN 978-1-4244-5788-5. doi: 10.1109/ICPADS.2009.90. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5395338. 24, 29

[258] Y. Zhao and N. Thomas. Efficient solutions of a PEPA model of a key distribution centre. *Performance Evaluation*, 67(8):740–756, Aug. 2010. ISSN 01665316. doi: 10.1016/j.peva.2009.07.005. URL http://dx.doi.org/10.1016/j.peva.2009.07.005. 24

[259] G. Zyba, G. M. Voelker, S. Ioannidis, and C. Diot. Dissemination in Opportunistic Mobile Ad-hoc Networks: the Power of the Crowd. In *30th IEEE Conference on Computer Communications (INFOCOM)*, pages 1179–1187. IEEE, 2011. 27, 30

Appendix A

Chapter 3

A.1 Lipschitz continuity of first-moment ODEs

For any GPEPA model, Lemma A.1.3 below verifies Lipschitz continuity of $\mathbf{f} : \mathbb{R}_+^{\mathcal{N}(G)} \to \mathbb{R}^{\mathcal{N}(G)}$ defined in Section 3.3. It uses the technical notion of the *structural depth* of a GPEPA model. This is the largest number of cooperations involving timed action type α, whose immediate effect can be seen by a fluid component enabling an α-action within some component group. The formal definition is given below before the lemma. Furthermore, we will require the *maximal local α-rate* for a timed action type α defined by $\mathcal{Q}_\alpha^{\max}(G) := \max_{(H,P)\in\mathcal{B}(G)}\{r_\alpha(P)\}$ and the *maximal local rate* defined by $\mathcal{Q}^{\max}(G) := \max_{(H,P)\in\mathcal{B}(G)}\{\sum_{\alpha\in\mathcal{A}} r_\alpha(P)\}$.

Definition A.1.1 (GPEPA :: Structural depth)**.** *For any GPEPA model G and timed action type $\alpha \in \mathcal{A}_t$, the structural depth of G with respect to α is $\mathcal{D}_\alpha(G)$, defined as follows.*

$$\mathcal{D}_\alpha(M_1 \underset{L}{\bowtie} M_2) := \begin{cases} 1 + \max\{\mathcal{D}_\alpha(M_1), \mathcal{D}_\alpha(M_2)\} & \text{if } \alpha \in L \\ \max\{\mathcal{D}_\alpha(M_1), \mathcal{D}_\alpha(M_2)\} & \text{if } \alpha \notin L \end{cases}$$

$$\mathcal{D}_\alpha(Y\{D\}) := 0$$

An auxiliary lemma, which follows, is required in the proof of Lemma A.1.3.

Lemma A.1.2. *Let $A \subseteq \mathbb{R}^n$ be convex and open. Let $\mathbf{g} : \bar{A} \to \mathbb{R}^n$ be a function continuous on $\bar{A}$ and differentiable on A. Assume also that for $i, j \in \{1, \ldots, n\}$:*

$$\sup_{\mathbf{x}\in A} \left| \frac{\partial g_i}{\partial x_j}(\mathbf{x}) \right| \leq \Lambda < \infty$$

Then $\mathbf{g}$ is Lipschitz continuous on $\bar{A}$ with a Lipschitz constant $n\Lambda$.

Proof. Let $\mathbf{x}, \mathbf{y} \in A$ be arbitrary, and define the function $\mathbf{G} : [0, 1] \to \mathbb{R}^n$ by $\mathbf{G}(t) := \mathbf{g}((1 - t)\mathbf{x} + t\mathbf{y})$. Now, by convexity of A, $\mathbf{G}$ is differentiable on $(0, 1)$ and we have for all $t \in (0, 1)$,

$\mathbf{G}'(t) = \mathbf{Dg}((1-t)\mathbf{x}+t\mathbf{y})\cdot(\mathbf{y}-\mathbf{x})$, by the chain rule, where the operator $\mathbf{D}$ gives the Jacobian matrix of the function. Then:

$$\begin{aligned}|G_i(1)-G_i(0)|^2 = \left[\int_0^1 \frac{dG_i}{\mathrm{d}t}(s)\,\mathrm{d}s\right]^2 &= \left[\int_0^1 \left(\frac{\partial g_i}{\partial x_1},\ldots,\frac{\partial g_i}{\partial x_n}\right)((1-s)\mathbf{x}+s\mathbf{y})\cdot(\mathbf{y}-\mathbf{x})\,\mathrm{d}s\right]^2 \\ &\leq \|\mathbf{y}-\mathbf{x}\|^2\left[\int_0^1 \left\|\left(\frac{\partial g_i}{\partial x_1},\ldots,\frac{\partial g_i}{\partial x_n}\right)((1-s)\mathbf{x}+s\mathbf{y})\right\|\,\mathrm{d}s\right]^2 \\ &\leq n\Lambda^2\|\mathbf{y}-\mathbf{x}\|^2\end{aligned}$$

by the Cauchy–Schwarz inequality. So:

$$\|\mathbf{g}(\mathbf{y})-\mathbf{g}(\mathbf{x})\| = \|\mathbf{G}(1)-\mathbf{G}(0)\| \leq n\Lambda\|\mathbf{y}-\mathbf{x}\|$$

as required. The extension to $\bar{A}$ is trivial by continuity of $\mathbf{g}$ and continuity of norms. □

Lemma A.1.3. *The function $\mathbf{f} : \mathbb{R}_+^{\mathcal{N}(G)} \to \mathbb{R}^{\mathcal{N}(G)}$ is Lipschitz continuous for any GPEPA model G. Furthermore, a Lipschitz constant (with respect to the usual Euclidean norm) is:*

$$\mathcal{K}(G) := 2\mathcal{N}(G)\sum_{\alpha\in\mathcal{A}_t}(\mathcal{D}_\alpha(G)+1)\mathcal{Q}_\alpha^{max}(G)$$

Proof. We see from Definition 3.3.1 that for $1 \leq k \leq \mathcal{N}(G)$ (corresponding to some $(H,P) \in \mathcal{B}(G)$) and $\mathbf{v} \in \mathbb{R}_+^{\mathcal{N}(G)}$ (corresponding to some $V \in \mathcal{B}(G) \to \mathbb{R}_+$):

$$f_k(\mathbf{v}) = \sum_{\alpha\in\mathcal{A}_t}\left[\left(\sum_{Q\in\mathcal{B}(G,H)} p_\alpha(Q,P)\,\mathcal{R}_\alpha(G,\mathbf{v},H,Q)\right) - \mathcal{R}_\alpha(G,\mathbf{v},H,P)\right]$$

For arbitrary $\alpha \in \mathcal{A}_t$, we focus now on a term, $\mathcal{R}_\alpha(G,\mathbf{v},H,Q)$ for $Q \in \mathcal{B}(G,H)$. It is a straightforward application of structural induction over Definition 3.2.4 to see that it has the following general form, for some $1 \leq i \leq \mathcal{N}(G)$ and $1 \leq D \leq \mathcal{D}_\alpha(G)$:

$$\mathcal{R}_\alpha(G,\mathbf{v},H,Q) = r_\alpha(Q)v_i \times \prod_{n=1}^{D}\frac{\min(a_n(\mathbf{v}),b_n(\mathbf{v}))}{a_n(\mathbf{v})} \tag{A.1}$$

where for any $1 \leq n \leq \mathcal{D}_\alpha(G)$:

$$a_n(\mathbf{v}) \geq r_\alpha(Q)v_i \times \prod_{m=1}^{n-1}\frac{\min(a_m(\mathbf{v}),b_m(\mathbf{v}))}{a_m(\mathbf{v})}$$

Now, the functions $a_n(\cdot)$ and $b_n(\cdot)$ are just instances of apparent rate (Definition 3.2.6). So they and their minimum, $\min(a_n(\cdot),b_n(\cdot))$, are all piecewise-linear on closed subsets of $\mathbb{R}_+^{\mathcal{N}(G)}$, each defined by a system of linear inequalities. These subsets thus form a covering of $\mathbb{R}_+^{\mathcal{N}(G)}$ by closed convex sets. Take an arbitrary such region, say $A \subseteq \mathbb{R}_+^{\mathcal{N}(G)}$. For $\mathbf{v} \in A$, some of the terms of the product in Equation A.1 will cancel and, re-ordering indices where necessary, for some $D' \leq \mathcal{D}_\alpha(G)$:

$$\mathcal{R}_\alpha(G,\mathbf{v},H,Q) = r_\alpha(Q)v_i \times \prod_{n=1}^{D'}\frac{b_n(\mathbf{v})}{a_n(\mathbf{v})}$$

and for any $1 \leq n \leq D'$, the following two inequalities hold:

$$\begin{aligned} a_n(\mathbf{v}) &\geq r_\alpha(P)v_i \times \prod_{m=1}^{n-1} \frac{b_m(\mathbf{v})}{a_m(\mathbf{v})} \\ a_n(\mathbf{v}) &\geq b_n(\mathbf{v}) \end{aligned} \tag{A.2}$$

On A, it is straightforward to see that $\mathcal{R}_\alpha(G, \cdot, H, Q)$ is continuous and, furthermore, on the interior of A, it is differentiable since the $a_n(\cdot)$ and $b_n(\cdot)$ are linear here. So for any $1 \leq j \leq \mathcal{N}(G)$ and $\mathbf{v} \in \text{int}(A)$:

$$\frac{\partial \mathcal{R}_\alpha(G, \cdot, H, Q)}{\partial v_j}(\mathbf{v}) = \begin{cases} r_\alpha(Q) \times \prod_{n=1}^{D'} \frac{b_n(\mathbf{v})}{a_n(\mathbf{v})} + r_\alpha(Q)v_i \times \frac{\partial}{\partial v_j}\left[\prod_{n=1}^{D'} \frac{b_n(\mathbf{v})}{a_n(\mathbf{v})}\right] & : j = i \\ r_\alpha(Q)v_i \times \frac{\partial}{\partial v_j}\left[\prod_{n=1}^{D'} \frac{b_n(\mathbf{v})}{a_n(\mathbf{v})}\right] & : j \neq i \end{cases}$$

Letting $F[l](\mathbf{v}) := \prod_{n=1}^{l} \frac{b_n(\mathbf{v})}{a_n(\mathbf{v})}$ for $0 \leq l \leq D'$, we have:

$$\frac{\partial F[l]}{\partial v_j}(\mathbf{v}) = \frac{F[l-1](\mathbf{v})}{a_l(\mathbf{v})}\frac{\partial b_l}{\partial v_j}(\mathbf{v}) - \frac{F[l](\mathbf{v})}{a_l(\mathbf{v})}\frac{\partial a_l}{\partial v_j}(\mathbf{v}) + \frac{b_l(\mathbf{v})}{a_l(\mathbf{v})}\frac{\partial F[l-1]}{\partial v_j}(\mathbf{v})$$

Applying the inequalities of Equation A.2, we obtain:

$$\left| r_\alpha(Q)v_i \times \frac{\partial F[l]}{\partial v_j}(\mathbf{v}) \right| \leq \left| \frac{\partial b_l}{\partial v_j}(\mathbf{v}) \right| + \left| \frac{\partial a_l}{\partial v_j}(\mathbf{v}) \right| + \left| r_\alpha(Q)v_i \times \frac{\partial F[l-1]}{\partial v_j}(\mathbf{v}) \right|$$

It is clear from the definition of apparent rate (Definition 3.2.6) that only one of $\left|\frac{\partial b_l}{\partial v_j}(\mathbf{v})\right|$ and $\left|\frac{\partial a_l}{\partial v_j}(\mathbf{v})\right|$ can be non-zero, and it is no greater than $r_\alpha(Q')$ for some fluid component, Q'. Thus by induction:

$$\sup_{\mathbf{v} \in \text{int}(A)} \left| \frac{\partial \mathcal{R}_\alpha(G, \cdot, H, Q)}{\partial v_j}(\mathbf{v}) \right| \leq (\mathcal{D}_\alpha(G) + 1)\mathcal{Q}_\alpha^{\max}(G)$$

Now considering all action types in the same way, we have:

$$\sup_{\mathbf{v} \in \text{int}(A)} \left| \frac{\partial f_k}{\partial v_j}(\mathbf{v}) \right| \leq 2 \sum_{\alpha \in \mathcal{A}_t} (\mathcal{D}_\alpha(G) + 1)\mathcal{Q}_\alpha^{\max}(G)$$

We now apply Lemma A.1.2 on the open convex set, $\text{int}(A)$ to show that $\mathbf{f}$ is Lipschitz continuous on A with a Lipschitz constant $\mathcal{K}(G) := 2\mathcal{N}(G) \sum_{\alpha \in \mathcal{A}_t} (\mathcal{D}_\alpha(G) + 1)\mathcal{Q}_\alpha^{\max}(G)$.

For general $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}_+^{\mathcal{N}(G)}$, consider the line connecting them: $(1-t)\mathbf{v}_1 + t\mathbf{v}_2$ for $t \in [0, 1]$. Let the closed convex sets making up the covering of $\mathbb{R}_+^{\mathcal{N}(G)}$ defined above be $\{A_i\}_{i=1}^N$. Assume the line between $\mathbf{v}_1$ and $\mathbf{v}_2$ intersects $k \geq 1$ of them, then re-ordering where necessary, there exist $\{t_j\}_{j=1}^{k+1}$ with $t_1 = 0$, $t_{k+1} = 1$ such that for each $1 \leq j < k$, $t_j < t_{j+1}$ and $(1-t)\mathbf{v}_1 + t\mathbf{v}_2 \in A_j$ for $t \in [t_j, t_{j+1}]$. Now write:

$$\begin{aligned} \|\mathbf{f}(\mathbf{v}_1) - \mathbf{f}(\mathbf{v}_2)\| &= \left\| \sum_{j=1}^{k} (\mathbf{f}((1-t_j)\mathbf{v}_1 + t_j\mathbf{v}_2) - \mathbf{f}((1-t_{j+1})\mathbf{v}_1 + t_{j+1}\mathbf{v}_2)) \right\| \\ &\leq \sum_{j=1}^{k} \|(\mathbf{f}((1-t_j)\mathbf{v}_1 + t_j\mathbf{v}_2) - \mathbf{f}((1-t_{j+1})\mathbf{v}_1 + t_{j+1}\mathbf{v}_2))\| \qquad \text{by the triangle inequality} \\ &\leq \mathcal{K}(G) \sum_{j=1}^{k} \|((1-t_j)\mathbf{v}_1 + t_j\mathbf{v}_2) - ((1-t_{j+1})\mathbf{v}_1 + t_{j+1}\mathbf{v}_2)\| \qquad \text{by Lipschitz continuity} \\ &= \mathcal{K}(G)\|\mathbf{v}_1 - \mathbf{v}_2\| \qquad \text{by colinearity} \end{aligned}$$

as required. □

A.2 Systems of ODEs

A.2.1 Second-moment ODEs for $\boldsymbol{CS}(n, m)$

Below is the complete system of 27 ODEs which approximate the second-order moments for the model $\boldsymbol{CS}(n, m)$ as derived according to Definition 3.4.5.

$$
\begin{aligned}
\dot{v}_{\boldsymbol{Ct}}(t) &= -v_{\boldsymbol{Ct}}(t)r_t + \min(v_{\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}}(t)r_d)\\
\dot{v}_{\boldsymbol{C}}(t) &= v_{\boldsymbol{Ct}}(t)r_t + v_{\boldsymbol{Cw}}(t)r_{tmt} - \min(v_{\boldsymbol{C}}(t)r_r, v_{\boldsymbol{S}}(t)r_r)\\
\dot{v}_{\boldsymbol{Ct}^2}(t) &= v_{\boldsymbol{Ct}}(t)r_t - 2v_{\boldsymbol{Ct}^2}(t)r_t + \min(v_{\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}}(t)r_d) + 2\min(v_{\boldsymbol{Cw}\cdot\boldsymbol{Ct}}(t)r_d, v_{\boldsymbol{Sg}\cdot\boldsymbol{Ct}}(t)r_d)\\
\dot{v}_{\boldsymbol{C}\cdot\boldsymbol{Ct}}(t) &= -v_{\boldsymbol{Ct}}(t)r_t + v_{\boldsymbol{Ct}^2}(t)r_t - v_{\boldsymbol{C}\cdot\boldsymbol{Ct}}(t)r_t + v_{\boldsymbol{Cw}\cdot\boldsymbol{Ct}}(t)r_{tmt}\\
&\quad + \min(v_{\boldsymbol{C}\cdot\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{C}\cdot\boldsymbol{Sg}}(t)r_d) - \min(v_{\boldsymbol{C}\cdot\boldsymbol{Ct}}(t)r_r, v_{\boldsymbol{S}\cdot\boldsymbol{Ct}}(t)r_r)\\
\dot{v}_{\boldsymbol{C}^2}(t) &= v_{\boldsymbol{Ct}}(t)r_t + 2v_{\boldsymbol{C}\cdot\boldsymbol{Ct}}(t)r_t + v_{\boldsymbol{Cw}}(t)r_{tmt} + 2v_{\boldsymbol{C}\cdot\boldsymbol{Cw}}(t)r_{tmt} + \min(v_{\boldsymbol{C}}(t)r_r, v_{\boldsymbol{S}}(t)r_r)\\
&\quad - 2\min(v_{\boldsymbol{C}^2}(t)r_r, v_{\boldsymbol{C}\cdot\boldsymbol{S}}(t)r_r)\\
\dot{v}_{\boldsymbol{S}\cdot\boldsymbol{Ct}}(t) &= -v_{\boldsymbol{S}\cdot\boldsymbol{Ct}}(t)r_t + v_{\boldsymbol{Ct}\cdot\boldsymbol{Sb}}(t)r_{rst} - v_{\boldsymbol{S}\cdot\boldsymbol{Ct}}(t)r_b + \min(v_{\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}}(t)r_d) + \min(v_{\boldsymbol{Cw}\cdot\boldsymbol{S}}(t)r_d, v_{\boldsymbol{Sg}\cdot\boldsymbol{S}}(t)r_d)\\
&\quad + \min(v_{\boldsymbol{Cw}\cdot\boldsymbol{Ct}}(t)r_d, v_{\boldsymbol{Sg}\cdot\boldsymbol{Ct}}(t)r_d) - \min(v_{\boldsymbol{C}\cdot\boldsymbol{Ct}}(t)r_r, v_{\boldsymbol{S}\cdot\boldsymbol{Ct}}(t)r_r)\\
\dot{v}_{\boldsymbol{C}\cdot\boldsymbol{S}}(t) &= v_{\boldsymbol{S}\cdot\boldsymbol{Ct}}(t)r_t + v_{\boldsymbol{Cw}\cdot\boldsymbol{S}}(t)r_{tmt} + v_{\boldsymbol{C}\cdot\boldsymbol{Sb}}(t)r_{rst} - v_{\boldsymbol{C}\cdot\boldsymbol{S}}(t)r_b + \min(v_{\boldsymbol{C}\cdot\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{C}\cdot\boldsymbol{Sg}}(t)r_d)\\
&\quad + \min(v_{\boldsymbol{C}}(t)r_r, v_{\boldsymbol{S}}(t)r_r) - \min(v_{\boldsymbol{C}\cdot\boldsymbol{S}}(t)r_r, v_{\boldsymbol{S}^2}(t)r_r) - \min(v_{\boldsymbol{C}^2}(t)r_r, v_{\boldsymbol{C}\cdot\boldsymbol{S}}(t)r_r)\\
\dot{v}_{\boldsymbol{Ct}\cdot\boldsymbol{Sb}}(t) &= -v_{\boldsymbol{Sb}\cdot\boldsymbol{Ct}}(t)r_t + v_{\boldsymbol{Sg}\cdot\boldsymbol{Ct}}(t)r_b - v_{\boldsymbol{Ct}\cdot\boldsymbol{Sb}}(t)r_{rst} + v_{\boldsymbol{S}\cdot\boldsymbol{Ct}}(t)r_b + \min(v_{\boldsymbol{Cw}\cdot\boldsymbol{Sb}}(t)r_d, v_{\boldsymbol{Sg}\cdot\boldsymbol{Sb}}(t)r_d)\\
\dot{v}_{\boldsymbol{C}\cdot\boldsymbol{Sb}}(t) &= v_{\boldsymbol{Sb}\cdot\boldsymbol{Ct}}(t)r_t + v_{\boldsymbol{Cw}\cdot\boldsymbol{Sb}}(t)r_{tmt} + v_{\boldsymbol{C}\cdot\boldsymbol{Sg}}(t)r_b - v_{\boldsymbol{C}\cdot\boldsymbol{Sb}}(t)r_{rst} + v_{\boldsymbol{C}\cdot\boldsymbol{S}}(t)r_b - \min(v_{\boldsymbol{C}\cdot\boldsymbol{Sb}}(t)r_r, v_{\boldsymbol{S}\cdot\boldsymbol{Sb}}(t)r_r)\\
\dot{v}_{\boldsymbol{Cw}\cdot\boldsymbol{Ct}}(t) &= -v_{\boldsymbol{Cw}\cdot\boldsymbol{Ct}}(t)r_t - v_{\boldsymbol{Cw}\cdot\boldsymbol{Ct}}(t)r_{tmt} - \min(v_{\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}}(t)r_d) + \min(v_{\boldsymbol{Cw}^2}(t)r_d, v_{\boldsymbol{Sg}\cdot\boldsymbol{Cw}}(t)r_d)\\
&\quad - \min(v_{\boldsymbol{Cw}\cdot\boldsymbol{Ct}}(t)r_d, v_{\boldsymbol{Sg}\cdot\boldsymbol{Ct}}(t)r_d) + \min(v_{\boldsymbol{C}\cdot\boldsymbol{Ct}}(t)r_r, v_{\boldsymbol{S}\cdot\boldsymbol{Ct}}(t)r_r)\\
\dot{v}_{\boldsymbol{C}\cdot\boldsymbol{Cw}}(t) &= v_{\boldsymbol{Cw}\cdot\boldsymbol{Ct}}(t)r_t - v_{\boldsymbol{Cw}}(t)r_{tmt} + v_{\boldsymbol{Cw}^2}(t)r_{tmt} - v_{\boldsymbol{C}\cdot\boldsymbol{Cw}}(t)r_{tmt} - \min(v_{\boldsymbol{C}\cdot\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{C}\cdot\boldsymbol{Sg}}(t)r_d)\\
&\quad - \min(v_{\boldsymbol{C}}(t)r_r, v_{\boldsymbol{S}}(t)r_r) - \min(v_{\boldsymbol{C}\cdot\boldsymbol{Cw}}(t)r_r, v_{\boldsymbol{Cw}\cdot\boldsymbol{S}}(t)r_r) + \min(v_{\boldsymbol{C}^2}(t)r_r, v_{\boldsymbol{C}\cdot\boldsymbol{S}}(t)r_r)\\
\dot{v}_{\boldsymbol{Sg}\cdot\boldsymbol{Ct}}(t) &= -v_{\boldsymbol{Sg}\cdot\boldsymbol{Ct}}(t)r_t - v_{\boldsymbol{Sg}\cdot\boldsymbol{Ct}}(t)r_b - \min(v_{\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}}(t)r_d) - \min(v_{\boldsymbol{Cw}\cdot\boldsymbol{Ct}}(t)r_d, v_{\boldsymbol{Sg}\cdot\boldsymbol{Ct}}(t)r_d)\\
&\quad + \min(v_{\boldsymbol{Sg}\cdot\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}^2}(t)r_d) + \min(v_{\boldsymbol{C}\cdot\boldsymbol{Ct}}(t)r_r, v_{\boldsymbol{S}\cdot\boldsymbol{Ct}}(t)r_r)\\
\dot{v}_{\boldsymbol{C}\cdot\boldsymbol{Sg}}(t) &= v_{\boldsymbol{Sg}\cdot\boldsymbol{Ct}}(t)r_t + v_{\boldsymbol{Sg}\cdot\boldsymbol{Cw}}(t)r_{tmt} - v_{\boldsymbol{C}\cdot\boldsymbol{Sg}}(t)r_b - \min(v_{\boldsymbol{C}\cdot\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{C}\cdot\boldsymbol{Sg}}(t)r_d) - \min(v_{\boldsymbol{C}}(t)r_r, v_{\boldsymbol{S}}(t)r_r)\\
&\quad + \min(v_{\boldsymbol{C}^2}(t)r_r, v_{\boldsymbol{C}\cdot\boldsymbol{S}}(t)r_r) - \min(v_{\boldsymbol{Sg}\cdot\boldsymbol{C}}(t)r_r, v_{\boldsymbol{Sg}\cdot\boldsymbol{S}}(t)r_r)\\
\dot{v}_{\boldsymbol{Cw}}(t) &= -v_{\boldsymbol{Cw}}(t)r_{tmt} - \min(v_{\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}}(t)r_d) + \min(v_{\boldsymbol{C}}(t)r_r, v_{\boldsymbol{S}}(t)r_r)\\
\dot{v}_{\boldsymbol{Cw}^2}(t) &= v_{\boldsymbol{Cw}}(t)r_{tmt} - 2v_{\boldsymbol{Cw}^2}(t)r_{tmt} + \min(v_{\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}}(t)r_d) - 2\min(v_{\boldsymbol{Cw}^2}(t)r_d, v_{\boldsymbol{Sg}\cdot\boldsymbol{Cw}}(t)r_d)\\
&\quad + \min(v_{\boldsymbol{C}}(t)r_r, v_{\boldsymbol{S}}(t)r_r) + 2\min(v_{\boldsymbol{C}\cdot\boldsymbol{Cw}}(t)r_r, v_{\boldsymbol{Cw}\cdot\boldsymbol{S}}(t)r_r)\\
\dot{v}_{\boldsymbol{Cw}\cdot\boldsymbol{S}}(t) &= -v_{\boldsymbol{Cw}\cdot\boldsymbol{S}}(t)r_{tmt} + v_{\boldsymbol{Cw}\cdot\boldsymbol{Sb}}(t)r_{rst} - v_{\boldsymbol{Cw}\cdot\boldsymbol{S}}(t)r_b - \min(v_{\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}}(t)r_d)\\
&\quad - \min(v_{\boldsymbol{Cw}\cdot\boldsymbol{S}}(t)r_d, v_{\boldsymbol{Sg}\cdot\boldsymbol{S}}(t)r_d) + \min(v_{\boldsymbol{Cw}^2}(t)r_d, v_{\boldsymbol{Sg}\cdot\boldsymbol{Cw}}(t)r_d) - \min(v_{\boldsymbol{C}}(t)r_r, v_{\boldsymbol{S}}(t)r_r)\\
&\quad + \min(v_{\boldsymbol{C}\cdot\boldsymbol{S}}(t)r_r, v_{\boldsymbol{S}^2}(t)r_r) - \min(v_{\boldsymbol{C}\cdot\boldsymbol{Cw}}(t)r_r, v_{\boldsymbol{Cw}\cdot\boldsymbol{S}}(t)r_r)\\
\dot{v}_{\boldsymbol{Cw}\cdot\boldsymbol{Sb}}(t) &= -v_{\boldsymbol{Cw}\cdot\boldsymbol{Sb}}(t)r_{tmt} + v_{\boldsymbol{Sg}\cdot\boldsymbol{Cw}}(t)r_b - v_{\boldsymbol{Cw}\cdot\boldsymbol{Sb}}(t)r_{rst} + v_{\boldsymbol{Cw}\cdot\boldsymbol{S}}(t)r_b - \min(v_{\boldsymbol{Cw}\cdot\boldsymbol{Sb}}(t)r_d, v_{\boldsymbol{Sg}\cdot\boldsymbol{Sb}}(t)r_d)\\
&\quad + \min(v_{\boldsymbol{C}\cdot\boldsymbol{Sb}}(t)r_r, v_{\boldsymbol{S}\cdot\boldsymbol{Sb}}(t)r_r)\\
\dot{v}_{\boldsymbol{Sg}\cdot\boldsymbol{Cw}}(t) &= -v_{\boldsymbol{Sg}\cdot\boldsymbol{Cw}}(t)r_{tmt} - v_{\boldsymbol{Sg}\cdot\boldsymbol{Cw}}(t)r_b + \min(v_{\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}}(t)r_d) - \min(v_{\boldsymbol{Cw}^2}(t)r_d, v_{\boldsymbol{Sg}\cdot\boldsymbol{Cw}}(t)r_d)\\
&\quad - \min(v_{\boldsymbol{Sg}\cdot\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}^2}(t)r_d) + \min(v_{\boldsymbol{C}}(t)r_r, v_{\boldsymbol{S}}(t)r_r) + \min(v_{\boldsymbol{C}\cdot\boldsymbol{Cw}}(t)r_r, v_{\boldsymbol{Cw}\cdot\boldsymbol{S}}(t)r_r)\\
&\quad + \min(v_{\boldsymbol{Sg}\cdot\boldsymbol{C}}(t)r_r, v_{\boldsymbol{Sg}\cdot\boldsymbol{S}}(t)r_r)\\
\dot{v}_{\boldsymbol{Sb}}(t) &= v_{\boldsymbol{Sg}}(t)r_b - v_{\boldsymbol{Sb}}(t)r_{rst} + v_{\boldsymbol{S}}(t)r_b\\
\dot{v}_{\boldsymbol{Sg}}(t) &= -v_{\boldsymbol{Sg}}(t)r_b - \min(v_{\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}}(t)r_d) + \min(v_{\boldsymbol{C}}(t)r_r, v_{\boldsymbol{S}}(t)r_r)\\
\dot{v}_{\boldsymbol{Sb}^2}(t) &= v_{\boldsymbol{Sg}}(t)r_b + 2v_{\boldsymbol{Sg}\cdot\boldsymbol{Sb}}(t)r_b + v_{\boldsymbol{Sb}}(t)r_{rst} - 2v_{\boldsymbol{Sb}^2}(t)r_{rst} + v_{\boldsymbol{S}}(t)r_b + 2v_{\boldsymbol{S}\cdot\boldsymbol{Sb}}(t)r_b
\end{aligned}
$$

$$
\begin{aligned}
\dot{v}_{\boldsymbol{Sg}\cdot\boldsymbol{Sb}}(t) = & -v_{\boldsymbol{Sg}}(t)r_b - v_{\boldsymbol{Sg}\cdot\boldsymbol{Sb}}(t)r_b + v_{\boldsymbol{Sg}^2}(t)r_b - v_{\boldsymbol{Sg}\cdot\boldsymbol{Sb}}(t)r_{rst} + v_{\boldsymbol{Sg}\cdot\boldsymbol{S}}(t)r_b \\
& - \min(v_{\boldsymbol{Cw}\cdot\boldsymbol{Sb}}(t)r_d, v_{\boldsymbol{Sg}\cdot\boldsymbol{Sb}}(t)r_d) + \min(v_{\boldsymbol{C}\cdot\boldsymbol{Sb}}(t)r_r, v_{\boldsymbol{S}\cdot\boldsymbol{Sb}}(t)r_r) \\
\dot{v}_{\boldsymbol{Sg}^2}(t) = & \, v_{\boldsymbol{Sg}}(t)r_b - 2v_{\boldsymbol{Sg}^2}(t)r_b + \min(v_{\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}}(t)r_d) - 2\min(v_{\boldsymbol{Sg}\cdot\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}^2}(t)r_d) \\
& + \min(v_{\boldsymbol{C}}(t)r_r, v_{\boldsymbol{S}}(t)r_r) + 2\min(v_{\boldsymbol{Sg}\cdot\boldsymbol{C}}(t)r_r, v_{\boldsymbol{Sg}\cdot\boldsymbol{S}}(t)r_r) \\
\dot{v}_{\boldsymbol{S}\cdot\boldsymbol{Sb}}(t) = & \, v_{\boldsymbol{Sg}\cdot\boldsymbol{S}}(t)r_b - v_{\boldsymbol{Sb}}(t)r_{rst} - v_{\boldsymbol{S}\cdot\boldsymbol{Sb}}(t)r_{rst} + v_{\boldsymbol{Sb}^2}(t)r_{rst} - v_{\boldsymbol{S}}(t)r_b + v_{\boldsymbol{S}^2}(t)r_b - v_{\boldsymbol{S}\cdot\boldsymbol{Sb}}(t)r_b \\
& + \min(v_{\boldsymbol{Cw}\cdot\boldsymbol{Sb}}(t)r_d, v_{\boldsymbol{Sg}\cdot\boldsymbol{Sb}}(t)r_d) - \min(v_{\boldsymbol{C}\cdot\boldsymbol{Sb}}(t)r_r, v_{\boldsymbol{S}\cdot\boldsymbol{Sb}}(t)r_r) \\
\dot{v}_{\boldsymbol{Sg}\cdot\boldsymbol{S}}(t) = & -v_{\boldsymbol{Sg}\cdot\boldsymbol{S}}(t)r_b + v_{\boldsymbol{Sg}\cdot\boldsymbol{Sb}}(t)r_{rst} - v_{\boldsymbol{Sg}\cdot\boldsymbol{S}}(t)r_b - \min(v_{\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}}(t)r_d) \\
& - \min(v_{\boldsymbol{Cw}\cdot\boldsymbol{S}}(t)r_d, v_{\boldsymbol{Sg}\cdot\boldsymbol{S}}(t)r_d) + \min(v_{\boldsymbol{Sg}\cdot\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}^2}(t)r_d) - \min(v_{\boldsymbol{C}}(t)r_r, v_{\boldsymbol{S}}(t)r_r) \\
& + \min(v_{\boldsymbol{C}\cdot\boldsymbol{S}}(t)r_r, v_{\boldsymbol{S}^2}(t)r_r) - \min(v_{\boldsymbol{Sg}\cdot\boldsymbol{C}}(t)r_r, v_{\boldsymbol{Sg}\cdot\boldsymbol{S}}(t)r_r) \\
\dot{v}_{\boldsymbol{S}}(t) = & \, v_{\boldsymbol{Sb}}(t)r_{rst} - v_{\boldsymbol{S}}(t)r_b + \min(v_{\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}}(t)r_d) - \min(v_{\boldsymbol{C}}(t)r_r, v_{\boldsymbol{S}}(t)r_r) \\
\dot{v}_{\boldsymbol{S}^2}(t) = & \, v_{\boldsymbol{Sb}}(t)r_{rst} + 2v_{\boldsymbol{S}\cdot\boldsymbol{Sb}}(t)r_{rst} + v_{\boldsymbol{S}}(t)r_b - 2v_{\boldsymbol{S}^2}(t)r_b + \min(v_{\boldsymbol{Cw}}(t)r_d, v_{\boldsymbol{Sg}}(t)r_d) \\
& + 2\min(v_{\boldsymbol{Cw}\cdot\boldsymbol{S}}(t)r_d, v_{\boldsymbol{Sg}\cdot\boldsymbol{S}}(t)r_d) + \min(v_{\boldsymbol{C}}(t)r_r, v_{\boldsymbol{S}}(t)r_r) - 2\min(v_{\boldsymbol{C}\cdot\boldsymbol{S}}(t)r_r, v_{\boldsymbol{S}^2}(t)r_r)
\end{aligned}
$$

Appendix B

Chapter 4

B.1 Systems of ODEs

B.1.1 First-moment ODEs for $CS'(n,m)$

$$
\begin{aligned}
\dot{v}_{\boldsymbol{Cw}'}(t) = & -v_{\boldsymbol{Cw}'}(t)r_{tmt} - \frac{v_{\boldsymbol{Cw}'}(t)}{v_{\boldsymbol{Cw}'}(t)+v_{\boldsymbol{Cw}}(t)}\min(v_{\boldsymbol{Sg}}(t)r_d, v_{\boldsymbol{Cw}'}(t)r_d + v_{\boldsymbol{Cw}}(t)r_d) \\
& + \frac{v_{\boldsymbol{C}'}(t)}{v_{\boldsymbol{C}'}(t)+v_{\boldsymbol{C}}(t)}\min(v_{\boldsymbol{S}}(t)r_r, v_{\boldsymbol{C}'}(t)r_r + v_{\boldsymbol{C}}(t)r_r) \\
\dot{v}_{\boldsymbol{C}'}(t) = & \, v_{\boldsymbol{Cw}'}(t)r_{tmt} + v_{\boldsymbol{Ct}}(t)r_t + v_{\boldsymbol{Ct}'}(t)r_t - \frac{v_{\boldsymbol{C}'}(t)}{v_{\boldsymbol{C}'}(t)+v_{\boldsymbol{C}}(t)}\min(v_{\boldsymbol{S}}(t)r_r, v_{\boldsymbol{C}'}(t)r_r + v_{\boldsymbol{C}}(t)r_r) \\
\dot{v}_{\boldsymbol{Ct}}(t) = & -v_{\boldsymbol{Ct}}(t)r_t + \frac{v_{\boldsymbol{Cw}}(t)}{v_{\boldsymbol{Cw}'}(t)+v_{\boldsymbol{Cw}}(t)}\min(v_{\boldsymbol{Sg}}(t)r_d, v_{\boldsymbol{Cw}'}(t)r_d + v_{\boldsymbol{Cw}}(t)r_d) \\
\dot{v}_{\boldsymbol{Cw}}(t) = & -v_{\boldsymbol{Cw}}(t)r_{tmt} - \frac{v_{\boldsymbol{Cw}}(t)}{v_{\boldsymbol{Cw}'}(t)+v_{\boldsymbol{Cw}}(t)}\min(v_{\boldsymbol{Sg}}(t)r_d, v_{\boldsymbol{Cw}'}(t)r_d + v_{\boldsymbol{Cw}}(t)r_d) \\
& + \frac{v_{\boldsymbol{C}}(t)}{v_{\boldsymbol{C}'}(t)+v_{\boldsymbol{C}}(t)}\min(v_{\boldsymbol{S}}(t)r_r, v_{\boldsymbol{C}'}(t)r_r + v_{\boldsymbol{C}}(t)r_r) \\
\dot{v}_{\boldsymbol{C}}(t) = & \, v_{\boldsymbol{Cw}}(t)r_{tmt} - \frac{v_{\boldsymbol{C}}(t)}{v_{\boldsymbol{C}'}(t)+v_{\boldsymbol{C}}(t)}\min(v_{\boldsymbol{S}}(t)r_r, v_{\boldsymbol{C}'}(t)r_r + v_{\boldsymbol{C}}(t)r_r) \\
\dot{v}_{\boldsymbol{Ct}'}(t) = & -v_{\boldsymbol{Ct}'}(t)r_t + \frac{v_{\boldsymbol{Cw}'}(t)}{v_{\boldsymbol{Cw}'}(t)+v_{\boldsymbol{Cw}}(t)}\min(v_{\boldsymbol{Sg}}(t)r_d, v_{\boldsymbol{Cw}'}(t)r_d + v_{\boldsymbol{Cw}}(t)r_d) \\
\dot{v}_{\boldsymbol{Sb}}(t) = & \, v_{\boldsymbol{Sg}}(t)r_b - v_{\boldsymbol{Sb}}(t)r_{rst} + v_{\boldsymbol{S}}(t)r_b \\
\dot{v}_{\boldsymbol{Sg}}(t) = & -v_{\boldsymbol{Sg}}(t)r_b - \min(v_{\boldsymbol{Sg}}(t)r_d, v_{\boldsymbol{Cw}'}(t)r_d + v_{\boldsymbol{Cw}}(t)r_d) + \min(v_{\boldsymbol{S}}(t)r_r, v_{\boldsymbol{C}'}(t)r_r + v_{\boldsymbol{C}}(t)r_r) \\
\dot{v}_{\boldsymbol{S}}(t) = & \, v_{\boldsymbol{Sb}}(t)r_{rst} - v_{\boldsymbol{S}}(t)r_b + \min(v_{\boldsymbol{Sg}}(t)r_d, v_{\boldsymbol{Cw}'}(t)r_d + v_{\boldsymbol{Cw}}(t)r_d) - \min(v_{\boldsymbol{S}}(t)r_r, v_{\boldsymbol{C}'}(t)r_r + v_{\boldsymbol{C}}(t)r_r)
\end{aligned}
$$

B.1.2 First-moment ODEs for $CS_2'(n,m)$

$$
\begin{aligned}
\dot{v}_{\widehat{\boldsymbol{Ct}}}(t) = & -v_{\widehat{\boldsymbol{Ct}}}(t)r_t + \frac{v_{\widehat{\boldsymbol{Cw}}}(t)r_d}{v_{\boldsymbol{Cw}'}(t)r_d + v_{\boldsymbol{Cw}}(t)r_d + v_{\widehat{\boldsymbol{Cw}}}(t)r_d}\min(v_{\boldsymbol{Sg}}(t)r_d, v_{\boldsymbol{Cw}'}(t)r_d + v_{\boldsymbol{Cw}}(t)r_d + v_{\widehat{\boldsymbol{Cw}}}(t)r_d) \\
\dot{v}_{\boldsymbol{C}'}(t) = & \, v_{\widehat{\boldsymbol{Ct}}}(t)r_t + v_{\boldsymbol{Cw}'}(t)r_{tmt} + v_{\boldsymbol{Ct}'}(t)r_t
\end{aligned}
$$

$$
\begin{aligned}
&- \frac{v_{\boldsymbol{C}'}(t)r_r}{v_{\boldsymbol{C}'}(t)r_r + v_{\widehat{\boldsymbol{C}}}(t)r_r + v_{\boldsymbol{C}}(t)r_r} \min(v_{\boldsymbol{S}}(t)r_r, v_{\boldsymbol{C}'}(t)r_r + v_{\widehat{\boldsymbol{C}}}(t)r_r + v_{\boldsymbol{C}}(t)r_r)\\
\dot{v}_{\boldsymbol{C}\boldsymbol{w}'}(t) = &- v_{\boldsymbol{C}\boldsymbol{w}'}(t)r_{tmt} + \frac{v_{\boldsymbol{C}'}(t)r_r}{v_{\boldsymbol{C}'}(t)r_r + v_{\widehat{\boldsymbol{C}}}(t)r_r + v_{\boldsymbol{C}}(t)r_r} \min(v_{\boldsymbol{S}}(t)r_r, v_{\boldsymbol{C}'}(t)r_r + v_{\widehat{\boldsymbol{C}}}(t)r_r + v_{\boldsymbol{C}}(t)r_r)\\
&- \frac{v_{\boldsymbol{C}\boldsymbol{w}'}(t)r_d}{v_{\boldsymbol{C}\boldsymbol{w}'}(t)r_d + v_{\boldsymbol{C}\boldsymbol{w}}(t)r_d + v_{\widehat{\boldsymbol{C}\boldsymbol{w}}}(t)r_d} \min(v_{\boldsymbol{S}\boldsymbol{g}}(t)r_d, v_{\boldsymbol{C}\boldsymbol{w}'}(t)r_d + v_{\boldsymbol{C}\boldsymbol{w}}(t)r_d + v_{\widehat{\boldsymbol{C}\boldsymbol{w}}}(t)r_d)\\
\dot{v}_{\boldsymbol{C}\boldsymbol{t}}(t) = &- v_{\boldsymbol{C}\boldsymbol{t}}(t)r_t + \frac{v_{\boldsymbol{C}\boldsymbol{w}}(t)r_d}{v_{\boldsymbol{C}\boldsymbol{w}'}(t)r_d + v_{\boldsymbol{C}\boldsymbol{w}}(t)r_d + v_{\widehat{\boldsymbol{C}\boldsymbol{w}}}(t)r_d} \min(v_{\boldsymbol{S}\boldsymbol{g}}(t)r_d, v_{\boldsymbol{C}\boldsymbol{w}'}(t)r_d + v_{\boldsymbol{C}\boldsymbol{w}}(t)r_d + v_{\widehat{\boldsymbol{C}\boldsymbol{w}}}(t)r_d)\\
\dot{v}_{\widehat{\boldsymbol{C}}}(t) = &\, v_{\boldsymbol{C}\boldsymbol{t}}(t)r_t + v_{\widehat{\boldsymbol{C}\boldsymbol{w}}}(t)r_{tmt} - \frac{v_{\widehat{\boldsymbol{C}}}(t)r_r}{v_{\boldsymbol{C}'}(t)r_r + v_{\widehat{\boldsymbol{C}}}(t)r_r + v_{\boldsymbol{C}}(t)r_r} \min(v_{\boldsymbol{S}}(t)r_r, v_{\boldsymbol{C}'}(t)r_r + v_{\widehat{\boldsymbol{C}}}(t)r_r + v_{\boldsymbol{C}}(t)r_r)\\
\dot{v}_{\boldsymbol{C}\boldsymbol{w}}(t) = &- v_{\boldsymbol{C}\boldsymbol{w}}(t)r_{tmt} + \frac{v_{\boldsymbol{C}}(t)r_r}{v_{\boldsymbol{C}'}(t)r_r + v_{\widehat{\boldsymbol{C}}}(t)r_r + v_{\boldsymbol{C}}(t)r_r} \min(v_{\boldsymbol{S}}(t)r_r, v_{\boldsymbol{C}'}(t)r_r + v_{\widehat{\boldsymbol{C}}}(t)r_r + v_{\boldsymbol{C}}(t)r_r)\\
&- \frac{v_{\boldsymbol{C}\boldsymbol{w}}(t)r_d}{v_{\boldsymbol{C}\boldsymbol{w}'}(t)r_d + v_{\boldsymbol{C}\boldsymbol{w}}(t)r_d + v_{\widehat{\boldsymbol{C}\boldsymbol{w}}}(t)r_d} \min(v_{\boldsymbol{S}\boldsymbol{g}}(t)r_d, v_{\boldsymbol{C}\boldsymbol{w}'}(t)r_d + v_{\boldsymbol{C}\boldsymbol{w}}(t)r_d + v_{\widehat{\boldsymbol{C}\boldsymbol{w}}}(t)r_d)\\
\dot{v}_{\boldsymbol{C}}(t) = &\, v_{\boldsymbol{C}\boldsymbol{w}}(t)r_{tmt} - \frac{v_{\boldsymbol{C}}(t)r_r}{v_{\boldsymbol{C}'}(t)r_r + v_{\widehat{\boldsymbol{C}}}(t)r_r + v_{\boldsymbol{C}}(t)r_r} \min(v_{\boldsymbol{S}}(t)r_r, v_{\boldsymbol{C}'}(t)r_r + v_{\widehat{\boldsymbol{C}}}(t)r_r + v_{\boldsymbol{C}}(t)r_r)\\
\dot{v}_{\widehat{\boldsymbol{C}\boldsymbol{w}}}(t) = &- v_{\widehat{\boldsymbol{C}\boldsymbol{w}}}(t)r_{tmt} + \frac{v_{\widehat{\boldsymbol{C}}}(t)r_r}{v_{\boldsymbol{C}'}(t)r_r + v_{\widehat{\boldsymbol{C}}}(t)r_r + v_{\boldsymbol{C}}(t)r_r} \min(v_{\boldsymbol{S}}(t)r_r, v_{\boldsymbol{C}'}(t)r_r + v_{\widehat{\boldsymbol{C}}}(t)r_r + v_{\boldsymbol{C}}(t)r_r)\\
&- \frac{v_{\widehat{\boldsymbol{C}\boldsymbol{w}}}(t)r_d}{v_{\boldsymbol{C}\boldsymbol{w}'}(t)r_d + v_{\boldsymbol{C}\boldsymbol{w}}(t)r_d + v_{\widehat{\boldsymbol{C}\boldsymbol{w}}}(t)r_d} \min(v_{\boldsymbol{S}\boldsymbol{g}}(t)r_d, v_{\boldsymbol{C}\boldsymbol{w}'}(t)r_d + v_{\boldsymbol{C}\boldsymbol{w}}(t)r_d + v_{\widehat{\boldsymbol{C}\boldsymbol{w}}}(t)r_d)\\
\dot{v}_{\boldsymbol{C}\boldsymbol{t}'}(t) = &- v_{\boldsymbol{C}\boldsymbol{t}'}(t)r_t + \frac{v_{\boldsymbol{C}\boldsymbol{w}'}(t)r_d}{v_{\boldsymbol{C}\boldsymbol{w}'}(t)r_d + v_{\boldsymbol{C}\boldsymbol{w}}(t)r_d + v_{\widehat{\boldsymbol{C}\boldsymbol{w}}}(t)r_d} \min(v_{\boldsymbol{S}\boldsymbol{g}}(t)r_d, v_{\boldsymbol{C}\boldsymbol{w}'}(t)r_d + v_{\boldsymbol{C}\boldsymbol{w}}(t)r_d + v_{\widehat{\boldsymbol{C}\boldsymbol{w}}}(t)r_d)\\
\dot{v}_{\boldsymbol{S}\boldsymbol{b}}(t) = &\, v_{\boldsymbol{S}\boldsymbol{g}}(t)r_b - v_{\boldsymbol{S}\boldsymbol{b}}(t)r_{rst} + v_{\boldsymbol{S}}(t)r_b\\
\dot{v}_{\boldsymbol{S}\boldsymbol{g}}(t) = &- v_{\boldsymbol{S}\boldsymbol{g}}(t)r_b + \min(v_{\boldsymbol{S}}(t)r_r, v_{\boldsymbol{C}'}(t)r_r + v_{\widehat{\boldsymbol{C}}}(t)r_r + v_{\boldsymbol{C}}(t)r_r) - \min(v_{\boldsymbol{S}\boldsymbol{g}}(t)r_d, v_{\boldsymbol{C}\boldsymbol{w}'}(t)r_d + v_{\boldsymbol{C}\boldsymbol{w}}(t)r_d + v_{\widehat{\boldsymbol{C}\boldsymbol{w}}}(t)r_d)\\
\dot{v}_{\boldsymbol{S}}(t) = &\, v_{\boldsymbol{S}\boldsymbol{b}}(t)r_{rst} - v_{\boldsymbol{S}}(t)r_b - \min(v_{\boldsymbol{S}}(t)r_r, v_{\boldsymbol{C}'}(t)r_r + v_{\widehat{\boldsymbol{C}}}(t)r_r + v_{\boldsymbol{C}}(t)r_r)\\
&+ \min(v_{\boldsymbol{S}\boldsymbol{g}}(t)r_d, v_{\boldsymbol{C}\boldsymbol{w}'}(t)r_d + v_{\boldsymbol{C}\boldsymbol{w}}(t)r_d + v_{\widehat{\boldsymbol{C}\boldsymbol{w}}}(t)r_d)
\end{aligned}
$$

Appendix C

Chapter 5

C.1 Immediate PEPA

C.1.1 Well-behaved *i*PEPA components

Formally, an iPEPA model component P is *free from immediate cycles* if $\mathcal{L}_P(P, \emptyset)$ evaluates to **true** where:

$$\begin{aligned}
\mathcal{L}_P(P_1 \underset{L}{\bowtie} P_2, A) &:= (\mathcal{L}_P(P_1, A) \wedge \mathcal{L}_P(P_2, A)) \vee (\mathcal{L}_P(P_1, A) \wedge \mathcal{L}_P(P_2, A \cup L)) \\
&\quad \vee (\mathcal{L}_P(P_2, A) \wedge \mathcal{L}_P(P_1, A \cup L)) \\
\mathcal{L}_P(P/L, A) &:= \mathcal{L}_P(P, A \setminus L) \\
\mathcal{L}_P(S, A) &:= \begin{cases} \textbf{true} & : \mathcal{L}_S(\boldsymbol{C}_S, S', A) \text{ is } \textbf{true} \text{ for all constant definitions} \\ & \quad \boldsymbol{C}_S \stackrel{def}{=} S' \text{ referenced by } S \\ \textbf{false} & : \text{otherwise} \end{cases}
\end{aligned}$$

The only way a cycle in the state space of a sequential iPEPA component can occur is through a recursive reference to a constant. For a sequential constant definition $\boldsymbol{C}_S \stackrel{def}{=} S$, the purpose of the function $\mathcal{L}_S(\boldsymbol{C}_S, S, A)$ is to evaluate to **true** if no immediate cycle consisting only of unsynchronised immediate actions can exist from $\boldsymbol{C}_S$ back to $\boldsymbol{C}_S$, in the context of synchronisation on the action types $A \subseteq \mathcal{A}$:

$$\begin{aligned}
\mathcal{L}_S(\boldsymbol{C}_S, (\alpha, r).S_1, A) &:= \textbf{true} \\
\mathcal{L}_S(\boldsymbol{C}_S, [\mathsf{a}, w].S_1, A) &:= \begin{cases} \textbf{true} & : \mathsf{a} \in A \\ \mathcal{L}_S(\boldsymbol{C}_S, S_1, A) & : \mathsf{a} \notin A \end{cases} \\
\mathcal{L}_S(\boldsymbol{C}_S, S_1 + S_2, A) &:= \mathcal{L}_S(\boldsymbol{C}_S, S_1, A) \wedge \mathcal{L}_S(\boldsymbol{C}_S, S_2, A) \\
\mathcal{L}_S(\boldsymbol{C}_S, \boldsymbol{C}'_S, A) &:= \begin{cases} \textbf{false} & : \boldsymbol{C}_S = \boldsymbol{C}'_S \\ \mathcal{L}_S(\boldsymbol{C}_S, S', A) & : \boldsymbol{C}_S \neq \boldsymbol{C}'_S \text{ where } \boldsymbol{C}'_S \stackrel{def}{=} S' \end{cases}
\end{aligned}$$

In addition to freedom from immediate cycles, we also wish to ensure that an iPEPA component has deterministic initial behaviour. That is, for any well-behaved iPEPA component, we require that there may be at most one path of immediate transitions emanating from the component's initial state.

C.2 Translation of non-standard regular expressions to DFAs

Let R_l^1 and R_l^2 be two regular expressions according to the grammar of Equation 5.4. 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, $\mathcal{A}(P)\backslash\{\tau\}$ where P is the fluid component to which the local probe will 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 $R_l^1 \varnothing R_l^2$.

C.2.1 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.2.2 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.2.3 Fail ($R_l^1 \oslash R_l^2$)

The DFA corresponding to $R_l^1 \oslash R_l^2$ 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 $a \in \Sigma$, $\delta((x, y), a) := (x, y)$;
- $q_0 := (q_0^1, q_0^2)$;
- $F := F^1 \times (Q^2 \setminus F^2)$.

Appendix D

Chapter 6

D.1 Proofs

D.1.1 Proof of Theorem 6.2.1

Proof. We begin by defining the following stochastic process:

$$\bar{\mathbf{M}}_{\bar{\mathtt{n}}}(t) := \bar{\mathbf{N}}_{\bar{\mathtt{n}}}(t) - \left(\bar{\mathtt{n}} + \int_0^t \mathbf{f}(\bar{\mathbf{N}}_{\bar{\mathtt{n}}}(s))\,\mathrm{d}s\right)$$

The term $\bar{\mathtt{n}} + \int_0^t \mathbf{f}(\bar{\mathbf{N}}_{\bar{\mathtt{n}}}(s))\,\mathrm{d}s$ can be thought of as a perturbed numerical integration of the approximating ODE $\dot{\bar{\mathbf{v}}}(t) = \mathbf{f}(\bar{\mathbf{v}}(t))$ driven by the CTMC $\bar{\mathbf{N}}_{\bar{\mathtt{n}}}(t)$. The stochastic process $\bar{\mathbf{M}}_{\bar{\mathtt{n}}}(t)$ thus measures the deviation of the CTMC from this quantity. We will proceed by showing that $\bar{\mathbf{M}}_{\bar{\mathtt{n}}}(t)$ is a martingale which will allow us to obtain strong bounds on its magnitude over $[0, T]$. Grönwall's inequality [97, Page 498] will then be applied to relate this perturbed numerical integration to the ODE solution $\bar{\mathtt{n}} + \int_0^t \mathbf{f}(\bar{\mathbf{v}}_{\bar{\mathtt{n}}}(s))\,\mathrm{d}s$ itself.

In order to show that $\bar{\mathbf{M}}_{\bar{\mathtt{n}}}(t)$ is a martingale, we are required to show for all $t, s \geq 0$, that $\mathbb{E}[\bar{\mathbf{M}}_{\bar{\mathtt{n}}}(s+t)\big|\,\mathcal{F}_s] = \bar{\mathbf{M}}_{\bar{\mathtt{n}}}(s)$ a.s., where $\mathcal{F}_s := \sigma(\bar{\mathbf{N}}_{\bar{\mathtt{n}}}(u) \;:\; u \leq s)$ is the natural filtration of $\bar{\mathbf{N}}_{\bar{\mathtt{n}}}(s)$, that is, the σ-algebra containing all of the information about the process up to time s. Note that:

$$\begin{aligned}\bar{\mathbf{M}}_{\bar{\mathtt{n}}}(s+t) &= \left[\bar{\mathbf{M}}_{\bar{\mathtt{n}}}(s) - \bar{\mathbf{N}}_{\bar{\mathtt{n}}}(s) + \left(\bar{\mathtt{n}} + \int_0^s \mathbf{f}(\bar{\mathbf{N}}_{\bar{\mathtt{n}}}(u))\,\mathrm{d}u\right)\right] + \bar{\mathbf{N}}_{\bar{\mathtt{n}}}(s+t) - \left(\bar{\mathtt{n}} + \int_0^{s+t} \mathbf{f}(\bar{\mathbf{N}}_{\bar{\mathtt{n}}}(u))\,\mathrm{d}u\right)\\ &= \bar{\mathbf{M}}_{\bar{\mathtt{n}}}(s) + \bar{\mathbf{N}}_{\bar{\mathtt{n}}}(s+t) - \left(\bar{\mathbf{N}}_{\bar{\mathtt{n}}}(s) + \int_s^{s+t} \mathbf{f}(\bar{\mathbf{N}}_{\bar{\mathtt{n}}}(u))\,\mathrm{d}u\right)\end{aligned}$$

so we have:

$$\mathbb{E}\left[\bar{\mathbf{M}}_{\bar{\mathtt{n}}}(s+t)\big|\,\mathcal{F}_s\right] = \bar{\mathbf{M}}_{\bar{\mathtt{n}}}(s) + \mathbb{E}\left[\bar{\mathbf{N}}_{\bar{\mathtt{n}}}(s+t) - \left(\bar{\mathbf{N}}_{\bar{\mathtt{n}}}(s) + \int_s^{s+t} \mathbf{f}(\bar{\mathbf{N}}_{\bar{\mathtt{n}}}(u))\,\mathrm{d}u\right)\middle|\,\mathcal{F}_s\right]$$

Therefore, in order to show the martingale property, we must show that the expectation term on the right is (a.s.) zero. However, conditioning on the value of $\bar{\mathbf{N}}_{\bar{\mathtt{n}}}(s)$ and exploiting the Markov property, we see that this is implied if we can show that $\mathbb{E}[\bar{\mathbf{M}}_{\bar{\mathtt{n}}}(t)] = 0$ for all $t \geq 0$.

Let the jump times of $\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(t)$ be $\{\tau_j\}_{j=1}^{\infty}$ with also $\tau_0 := 0$, that is, $\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(\tau_j-) \neq \bar{\mathbf{N}}_{\bar{\mathbf{n}}}(\tau_j)$ for $j > 0$. Now consider $\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)$ stopped at τ_1, i.e. $\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t \wedge \tau_1)$.[1] Letting $q(\bar{\mathbf{n}})$ be the sum of all of the rates of all outgoing transitions from state $\bar{\mathbf{n}}$, we have:

$$\begin{aligned}
\mathbb{E}[\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t \wedge \tau_1)] &= \mathbb{E}[\mathbf{1}_{\{0 \leq t \leq \tau_1\}} \bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)] + \mathbb{E}[\mathbf{1}_{\{t > \tau_1\}} \bar{\mathbf{M}}_{\bar{\mathbf{n}}}(\tau_1)] \\
&= -\mathbb{E}[\mathbf{1}_{\{0 \leq t \leq \tau_1\}} t \mathbf{f}(\bar{\mathbf{n}})] + \mathbb{E}[\mathbf{1}_{\{t > \tau_1\}} (\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(\tau_1) - \bar{\mathbf{n}} - \tau_1 \mathbf{f}(\bar{\mathbf{n}}))] \\
&= -\exp(-q(\bar{\mathbf{n}})t) t \mathbf{f}(\bar{\mathbf{n}}) + (1 - \exp(-q(\bar{\mathbf{n}})t)) \frac{\mathbf{f}(\bar{\mathbf{n}})}{q(\bar{\mathbf{n}})} - \frac{\mathbf{f}(\bar{\mathbf{n}})}{q(\bar{\mathbf{n}})} [1 - \exp(-q(\bar{\mathbf{n}})t)(1 + q(\bar{\mathbf{n}})t)] \\
&= 0
\end{aligned}$$

Noting further that:

$$\begin{aligned}
&\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t \wedge \tau_{j+1}) - \bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t \wedge \tau_j) = \\
&\quad -\mathbf{1}_{\{\tau_j \leq t < \tau_{j+1}\}} (t - \tau_j) \mathbf{f}(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(\tau_j)) + \mathbf{1}_{\{t \geq \tau_{j+1}\}} (\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(\tau_{j+1}) - \bar{\mathbf{N}}_{\bar{\mathbf{n}}}(\tau_j) - (\tau_{j+1} - \tau_j) \mathbf{f}(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(\tau_j)))
\end{aligned}$$

for all $j \geq 0$, we may repeat the argument inductively, conditioning on possible values of $(\tau_j, \bar{\mathbf{N}}_{\bar{\mathbf{n}}}(\tau_j))$, to see that $\mathbb{E}[\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t \wedge \tau_j)] = 0$ for all $j \geq 0$. Therefore, we may write for all $j \geq 0$:

$$\mathbb{E}[\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)] = \mathbb{E}[\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t \wedge \tau_j)] + \mathbb{E}[\mathbf{1}_{\{t \geq \tau_j\}} (\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t) - \bar{\mathbf{M}}_{\bar{\mathbf{n}}}(\tau_j))] = \mathbb{E}[\mathbf{1}_{\{t \geq \tau_j\}} (\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t) - \bar{\mathbf{M}}_{\bar{\mathbf{n}}}(\tau_j))]$$

and also in the limit:

$$\mathbb{E}[\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)] = \lim_{j \to \infty} \mathbb{E}[\mathbf{1}_{\{t \geq \tau_j\}} (\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t) - \bar{\mathbf{M}}_{\bar{\mathbf{n}}}(\tau_j))] = \lim_{j \to \infty} \mathbb{E}[\mathbf{1}_{\{t \geq \tau_j\}} \bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)] - \lim_{j \to \infty} \mathbb{E}[\mathbf{1}_{\{t \geq \tau_j\}} \bar{\mathbf{M}}_{\bar{\mathbf{n}}}(\tau_j)] \tag{D.1}$$

We can bound τ_j in distribution by an Erlang random variable, say μ_j, with parameters j and $\mathcal{S}(G)\mathcal{Q}^{\max}(G)$ (the finite maximum jump rate of $\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(t)$), that is, $\mathbb{P}\{\tau_j \leq t\} \leq \mathbb{P}\{\mu_j \leq t\}$. So clearly $\lim_{j \to \infty} \mathbb{E}[\mathbf{1}_{\{t \geq \tau_j\}}] = 0$. Then the first term of Equation D.1 is zero by monotone convergence and the second is zero by the Cauchy-Schwarz inequality. Thus $\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)$ is a martingale as required.

Recall from Lemma A.1.3 that a Lipschitz constant of the function $\mathbf{f}$ is $\mathcal{K}(G)$. Now note that by definition and the triangle inequality, we have for all $t \in \mathbb{R}_+$:

$$\begin{aligned}
\|\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(t) - \bar{\mathbf{v}}_{\bar{\mathbf{n}}}(t)\| &= \left\| \bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t) + \int_0^t \mathbf{f}(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(s)) \, \mathrm{d}s - \int_0^t \mathbf{f}(\bar{\mathbf{v}}_{\bar{\mathbf{n}}}(s)) \, \mathrm{d}s \right\| \\
&\leq \|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)\| + \int_0^t \|\mathbf{f}(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(s)) - \mathbf{f}(\bar{\mathbf{v}}_{\bar{\mathbf{n}}}(s))\| \, \mathrm{d}s \\
&\leq \|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)\| + \mathcal{K}(G) \int_0^t \|\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(s) - \bar{\mathbf{v}}_{\bar{\mathbf{n}}}(s)\| \, \mathrm{d}s
\end{aligned} \tag{D.2}$$

Let $\delta := \epsilon \exp(-\mathcal{K}(G)T)$ and define the event $A := \left\{ \sup_{t \in [0,T]} \|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)\| > \delta \right\}$. On the complement of this event, we may apply Grönwall's inequality[2] to the function, $e(t) := \|\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(t) - \bar{\mathbf{v}}_{\bar{\mathbf{n}}}(t)\|$, to obtain, $\sup_{t \in [0,T]} \|\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(t) - \bar{\mathbf{v}}_{\bar{\mathbf{n}}}(t)\| \leq \epsilon$. Thus we have shown the following result:

$$\mathbb{P}\left\{ \sup_{t \in [0,T]} \|\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(t) - \bar{\mathbf{v}}_{\bar{\mathbf{n}}}(t)\| > \epsilon \right\} \leq \mathbb{P}(A)$$

[1] We use the shorthand $a \wedge b := \min(a, b)$.

[2] Grönwall's inequality says that for any real-valued integrable function g on $[0, S]$, the inequality $g(s) \leq C + D \int_0^s g(u) \, \mathrm{d}u$ for all $s \in [0, S]$ implies $g(S) \leq C \exp(DS)$ (see e.g. Ethier and Kurtz [97, Page 498]).

So it remains to bound $\mathbb{P}(A)$. Note that for any $k \geq 0$, by conditioning on the two possibilities $\{\tau_k < T\}$ or $\{\tau_k \geq T\}$, we may obtain:

$$\left\{ \sup_{t\in[0,T]} \|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)\| > \delta \right\} = \left\{ \sup_{t\in[0,T]} \|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)\|^2 > \delta^2 \right\} = \{\tau_k < T\} \cup \left\{ \sup_{t\in[0,T]} \|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t \wedge \tau_k)\|^2 > \delta^2 \right\}$$

where $\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t \wedge \tau_k)$ is $\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)$ stopped at τ_k, and is also a martingale by the optional stopping theorem.[3] So we can apply Doob's L^2-martingale inequality [e.g. 210, Page 177] to obtain:

$$\mathbb{P}\left\{ \sup_{t\in[0,T]} \|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t \wedge \tau_k)\|^2 > \delta^2 \right\} \leq \delta^{-2}\mathbb{E}\left[\sup_{t\in[0,T]} \|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t \wedge \tau_k)\|^2 \right] \leq 4\delta^{-2}\mathbb{E}\left[\|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(T \wedge \tau_k)\|^2 \right]$$

Thus we have, for any $k \geq 0$, $\mathbb{P}\left\{ \sup_{t\in[0,T]} \|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)\| > \delta \right\} \leq \mathbb{P}\{\tau_k < T\} + 4\delta^{-2}\mathbb{E}\left[\|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(T \wedge \tau_k)\|^2 \right]$.

Let $k = \mathcal{S}(G)(\lceil \mathcal{Q}^{\max}(G)T \rceil + 1)$. Now τ_k is bounded below in distribution by the sum of k independent and identically exponentially-distributed random variables each with parameter the maximal jump rate of $\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(t)$, $\mathcal{S}(G)\mathcal{Q}^{\max}(G)$. Denote this Erlang random variable by μ_k. So we have:

$$\mathbb{P}\{\tau_k < T\} \leq \mathbb{P}\{\mu_k < T\} = \frac{\gamma(\mathcal{S}(G)(\lceil \mathcal{Q}^{\max}(G)T \rceil + 1), \mathcal{S}(G)\mathcal{Q}^{\max}(G)T)}{(\mathcal{S}(G)(\lceil \mathcal{Q}^{\max}(G)T \rceil + 1) - 1)!}$$

where γ is the lower incomplete gamma function.

Now to bound $\mathbb{E}\left[\|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(T \wedge \tau_k)\|^2 \right]$, note that by considering the possibilities for the jump times between which T lies, and observing that $\mathbb{E}[\mathbf{1}_A X] \leq \mathbb{P}(A)\mathbb{E}[X]$ for any events A and random variables X:

$$\begin{aligned} \mathbb{E}\left[\|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(T \wedge \tau_k)\|^2 \right] &= \mathbb{E}[\mathbf{1}_{\{T \geq \tau_k\}} \|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(\tau_k)\|^2] + \sum_{i=0}^{k-1} \mathbb{E}[\mathbf{1}_{\{\tau_i \leq T < \tau_{i+1}\}} \|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(\tau_i)\|^2] \\ &\leq \mathbb{P}\{T \geq \tau_k\}\mathbb{E}[\|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(\tau_k)\|^2] + \sum_{i=0}^{k-1} \mathbb{P}\{\tau_i \leq T < \tau_{i+1}\}\mathbb{E}[\|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(\tau_i)\|^2] \end{aligned} \tag{D.3}$$

For any $i \geq 0$, $\mathbb{E}[\|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(\tau_i)\|^2] \leq \sum_{j=0}^{i-1} \mathbb{E}[\|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(\tau_{j+1}) - \bar{\mathbf{M}}_{\bar{\mathbf{n}}}(\tau_j)\|^2]$ by the triangle inequality, and for any $j \geq 0$:

$$\begin{aligned} \mathbb{E}[\|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(\tau_{j+1}) - \bar{\mathbf{M}}_{\bar{\mathbf{n}}}(\tau_j)\|^2] = \sum_{l=1}^{\mathcal{N}(G)} \mathbb{E}[&(\bar{N}_{\bar{\mathbf{n}},l}(\tau_{j+1}) - \bar{N}_{\bar{\mathbf{n}},l}(\tau_j))^2 + (\tau_{j+1} - \tau_j)^2 f_l^2(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(\tau_j)) \\ &- 2(\bar{N}_{\bar{\mathbf{n}},l}(\tau_{j+1}) - \bar{N}_{\bar{\mathbf{n}},l}(\tau_j))(\tau_{j+1} - \tau_j) f_l(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(\tau_j))] \end{aligned} \tag{D.4}$$

by expanding the norm component wise. Then, for any CTMC state $\bar{\mathbf{n}}'$, we have:

$$\begin{aligned} &\mathbb{E}[2(\bar{N}_{\bar{\mathbf{n}},l}(\tau_{j+1}) - \bar{N}_{\bar{\mathbf{n}},l}(\tau_j))(\tau_{j+1} - \tau_j) f_l(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(\tau_j)) \mid \bar{\mathbf{N}}_{\bar{\mathbf{n}}}(\tau_j) = \bar{\mathbf{n}}'] \\ &\quad = 2f_l^2(\bar{\mathbf{n}}')/q^2(\bar{\mathbf{n}}') = \mathbb{E}[(\tau_{j+1} - \tau_j)^2 f_l^2(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(\tau_j)) \mid \bar{\mathbf{N}}_{\bar{\mathbf{n}}}(\tau_j) = \bar{\mathbf{n}}'] \end{aligned}$$

[3] See e.g. Rogers and Williams [210, Chapter 2, Section 5] or any standard text on continuous-time stochastic processes for the optional stopping theorem and martingale inequalities.

which is seen to be true by noting that on the event $\{\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(\tau_j) = \bar{\mathbf{n}}'\}$, $\tau_{j+1} - \tau_j$ is an exponential random variable with rate parameter $q(\bar{\mathbf{n}}')$, and $\bar{N}_{\bar{\mathbf{n}},l}(\tau_{j+1})$ is a random variable conditionally-independent of $\tau_{j+1} - \tau_j$ whose conditional expectation is $\bar{\mathbf{n}}' + f_l(\bar{\mathbf{n}}')/q(\bar{\mathbf{n}}')$.

Thus, combining Equations D.3 and D.4, we obtain:

$$\mathbb{E}\left[\|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(T \wedge \tau_k)\|^2\right] \leq k \sum_{l=1}^{\mathcal{N}(G)} \mathbb{E}[(\bar{N}_{\bar{\mathbf{n}},l}(\tau_{k+1}) - \bar{N}_{\bar{\mathbf{n}},l}(\tau_k))^2] \leq \frac{k\mathcal{N}(G)}{\mathcal{S}^2(G)} = \frac{(\lceil \mathcal{Q}^{\max}(G)T \rceil + 1)\mathcal{N}(G)}{\mathcal{S}(G)}$$

since each component count changes by at most one component at each jump time (Theorem 3.2.7). Thus we have the required bound:

$$\mathbb{P}(A) \leq \frac{\gamma(\mathcal{S}(G)(\lceil \mathcal{Q}^{\max}(G)T \rceil + 1), \mathcal{S}(G)\mathcal{Q}^{\max}(G)T)}{(\mathcal{S}(G)(\lceil \mathcal{Q}^{\max}(G)T \rceil + 1) - 1)!} + \frac{4(\lceil \mathcal{Q}^{\max}(G)T \rceil + 1)\mathcal{N}(G)\exp(2\mathcal{K}(G)T)}{\mathcal{S}(G)\epsilon^2}$$

□

D.1.2 Proof of Theorem 6.2.2

Proof. Before we begin, we decompose the function $\mathbf{f} = (+1) \times \mathbf{q}^+ + (-1) \times \mathbf{q}^-$ such that $\mathbf{q}^-$ specifies *the rate at which each component count is decremented* and $\mathbf{q}^+$ specifies *the rate at which each component count is incremented*, that is, the components of these functions are given by Item 1 and Item 2 of Theorem 3.2.7, respectively, summed over all timed action types.

Since $\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)$ is a (vector) martingale, its components and the negated versions of them $\pm\bar{M}_{\bar{\mathbf{n}},l}(t)$ for $l = 1, \ldots, \mathcal{N}(G)$ are martingales. Theorem 8.4 of Darling and Norris [85] can then be used to show that the following processes are also martingales:

$$\bar{Z}_{\bar{\mathbf{n}},l}(t,\theta) := \exp\left(\theta\bar{M}_{\bar{\mathbf{n}},l}(t) - \int_0^t \sigma_l(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(s),\theta)\,\mathrm{d}s\right)$$

for any $\theta \in \mathbb{R}$. We will see later that the extra parameter θ is useful in controlling the bound we can obtain. The function σ_l is defined by:

$$\sigma_l(\mathbf{x},\theta) := q_l^+(\mathcal{S}(G)\mathbf{x})h(\theta/\mathcal{S}(G)) + q_l^-(\mathcal{S}(G)\mathbf{x})h(-\theta/\mathcal{S}(G))$$

where $h(x) := \exp(x) - 1 - x$.

Recall from the proof of Theorem 6.2.1 that the stochastic process $\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)$ essentially represents the deviation of the CTMC from a perturbed numerical integration of the approximating ODE $\dot{\bar{\mathbf{v}}}(t) = \mathbf{f}(\bar{\mathbf{v}}(t))$. This idea was then used to obtain bounds on the deviation of the ODE solution itself from the CTMC. The idea of the constructions $\bar{Z}_{\bar{\mathbf{n}},l}(t,\theta)$ is that by taking the exponential of $\bar{M}_{\bar{\mathbf{n}},l}(t)$, in some sense, we magnify the deviation from the ODE and thus we might expect to obtain tighter bounds. The term $-\int_0^t \sigma_l(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(s),\theta)\,\mathrm{d}s$ is a correction factor which is required to maintain the martingale property of $\bar{Z}_{\bar{\mathbf{n}},l}(t,\theta)$, which allows us to use martingale theorems (like before) to obtain strong bounds on the magnitude of $\bar{Z}_{\bar{\mathbf{n}},l}(t,\theta)$.

Now define the bounded stopping time $T_B := \min(\inf\{t \geq 0 : \theta\bar{M}_{\bar{\mathbf{n}},l}(t) > B\}, T)$ and the event:

$$\Omega_l^\theta := \left\{\sup_{t\in[0,T]} \theta\bar{M}_{\bar{\mathbf{n}},l}(t) > B\right\} \cap \left\{\int_0^T \sigma_l(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(s),\theta)\,\mathrm{d}s \leq C\right\}$$

Then we have $\mathbb{E}[\bar{Z}_{\bar{\mathbf{n}},l}(T_B,\theta)] \geq \mathbb{P}(\Omega_l^\theta)\exp(B-C)$. By the optional stopping theorem and since $\bar{Z}_{\bar{\mathbf{n}},l}(t,\theta)$ is a martingale, we have $\mathbb{E}[\bar{Z}_{\bar{\mathbf{n}},l}(T_B,\theta)] = 1$ and thus:

$$\mathbb{P}(\Omega_l^\theta) \leq \exp(C-B) \tag{D.5}$$

Let $\delta := \epsilon\exp\left(-\sqrt{\mathcal{N}(G)}\mathcal{K}(G)T\right)$ and define the event $A := \left\{\sup_{t\in[0,T]}\|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)\|_\infty > \delta\right\}$. We will now proceed in a similar manner to the proof of Theorem 6.2.1 in that we can use Grönwall's inequality to see that:

$$\mathbb{P}\left\{\sup_{t\in[0,T]}\|\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(t) - \bar{\mathbf{v}}_{\bar{\mathbf{n}}}(t)\|_\infty > \epsilon\right\} \leq \mathbb{P}(A)$$

Note however that we are now using the *maximum norm* given by $\|\mathbf{x}\|_\infty := \max\{|x_1|,\ldots,|x_n|\}$ for $\mathbf{x}\in\mathbb{R}^n$, hence the extra $\sqrt{\mathcal{N}(G)}$ term in the definition of δ.[4] Now define:

$$\sigma^m(\mathbf{x},\theta) := \max_{1\leq l\leq\mathcal{N}(G)}\left\{q_l^+(\mathcal{S}(G)\mathbf{x})h(|\theta|/\mathcal{S}(G)) + q_l^-(\mathcal{S}(G)\mathbf{x})h(|\theta|/\mathcal{S}(G))\right\}$$

and note that $\sigma_l(\mathbf{x},\theta) \leq \sigma^m(\mathbf{x},\theta)$ for all l since $h(x)\geq h(-x)$ for all $x\geq 0$.[5] Then we have:

$$\begin{aligned}
\mathbb{P}(A) \leq & \sum_{l=1}^{\mathcal{N}(G)}\mathbb{P}\left(\left\{\sup_{t\in[0,T]}\theta\bar{M}_{\bar{\mathbf{n}},l}(t) > \theta\delta\right\}\cap\left\{\int_0^T\sigma^m(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(s),\theta)\,\mathrm{d}s\leq C\right\}\right)\\
&+\sum_{l=1}^{\mathcal{N}(G)}\mathbb{P}\left(\left\{\sup_{t\in[0,T]}-\theta\bar{M}_{\bar{\mathbf{n}},l}(t) > \theta\delta\right\}\cap\left\{\int_0^T\sigma^m(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(s),\theta)\,\mathrm{d}s\leq C\right\}\right)\\
&+\mathbb{P}\left(\left\{\int_0^T\sigma^m(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(s),\theta)\,\mathrm{d}s > C\right\}\right)\\
\leq & \sum_{l=1}^{\mathcal{N}(G)}\mathbb{P}\left(\left\{\sup_{t\in[0,T]}\theta\bar{M}_{\bar{\mathbf{n}},l}(t) > \theta\delta\right\}\cap\left\{\int_0^T\sigma_l(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(s),\theta)\,\mathrm{d}s\leq C\right\}\right)\\
&+\sum_{l=1}^{\mathcal{N}(G)}\mathbb{P}\left(\left\{\sup_{t\in[0,T]}-\theta\bar{M}_{\bar{\mathbf{n}},l}(t) > \theta\delta\right\}\cap\left\{\int_0^T\sigma_l(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(s),-\theta)\,\mathrm{d}s\leq C\right\}\right)\\
&+\mathbb{P}\left(\left\{\int_0^T\sigma^m(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(s),\theta)\,\mathrm{d}s > C\right\}\right)
\end{aligned}$$

This follows by noting that $\|\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)\|_\infty > \delta$ for some $t\in[0,T]$ only if $\bar{M}_{\bar{\mathbf{n}},l}(t) > \delta$ or $-\bar{M}_{\bar{\mathbf{n}},l}(t) > \delta$ for some $l\in\{1,\ldots,\mathcal{N}(G)\}$. Furthermore, in order to exploit Equation D.5, we condition also on the event $\left\{\int_0^T\sigma^m(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(s),\theta)\,\mathrm{d}s\leq C\right\}$. Indeed, applying Equation D.5 to the first $2\mathcal{N}(G)$ terms in the above then yields:

$$\mathbb{P}(A) \leq 2\mathcal{N}(G)\exp(C-\theta\delta) + \mathbb{P}\left(\left\{\int_0^T\sigma^m(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(s),\theta)\,\mathrm{d}s > C\right\}\right) \tag{D.6}$$

Choose now:

$$C := T\mathcal{Q}^{\max}(G)\mathcal{S}(G)(\exp(|\theta|/\mathcal{S}(G)) - 1 - |\theta|/\mathcal{S}(G))$$

[4] Since $\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\| \leq \sqrt{n}\|\mathbf{x}\|_\infty$ for $\mathbf{x}\in\mathbb{R}^n$, if K is a Lipschitz constant for $\|\cdot\|$, then $\sqrt{n}K$ is one for $\|\cdot\|_\infty$.

[5] That $h(x)\geq h(-x)$ for all $x\geq 0$ can be verified easily by expanding in Taylor series.

and it is clear that $\int_0^T \sigma^m(\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(s),\theta)\,\mathrm{d}s \leq C$ surely so that the final term in the right-hand side of Equation D.6 is zero. This gives finally the bound:

$$\mathbb{P}\left\{\sup_{t\in[0,T]} \|\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(t) - \bar{\mathbf{v}}_{\bar{\mathbf{n}}}(t)\|_\infty > \epsilon\right\} \leq$$
$$2\mathcal{N}(G)\exp\left[T\mathcal{Q}^{\max}(G)\mathcal{S}(G)(\exp(|\theta|/\mathcal{S}(G)) - 1 - |\theta|/\mathcal{S}(G)) - \theta\delta\right] \tag{D.7}$$

which holds for all $\theta \in \mathbb{R}$. The best possible such bound is obtained by minimising $m(\theta) := T\mathcal{Q}^{\max}(G)\mathcal{S}(G)(\exp[\theta/\mathcal{S}(G)\} - 1 - \theta/\mathcal{S}(G)] - \theta\delta$ with respect to $\theta \geq 0$. In fact, this is a convex function[6] and has a local minimum at $\theta^* := \mathcal{S}(G)\log(\delta/(T\mathcal{Q}^{\max}(G)) + 1) \geq 0$. The optimal bound is then obtained by substituting θ^* into the right-hand side of Equation D.7, which gives the bound of Equation 6.2.

To obtain the bound of Equation 6.3, set $\theta = l\mathcal{S}(G)$ for $l \in \mathbb{R}_+$. Then Equation D.7 becomes:

$$\mathbb{P}\left\{\sup_{t\in[0,T]} \|\bar{\mathbf{N}}_{\bar{\mathbf{n}}}(t) - \bar{\mathbf{v}}_{\bar{\mathbf{n}}}(t)\|_\infty > \epsilon\right\} \leq$$
$$2\mathcal{N}(G)\exp\left[\mathcal{S}(G)(T\mathcal{Q}^{\max}(G)(\exp(l) - 1 - l) - l\delta)\right]$$

We need $T\mathcal{Q}^{\max}(G)(\exp(l) - 1 - l) - l\delta < 0$ by an appropriate choice of l independent of $\mathcal{S}(G)$. Choose $l > 0$ such that $l\exp(l) < \frac{2\delta}{T\mathcal{Q}^{\max}(G)}$, then the result follows since $\exp(x) - x - 1 \leq \frac{1}{2}x^2\exp(x)$ for $x \in \mathbb{R}_+$. □

D.1.3 Proof of Corollary 6.2.3

Proof. First we note that $\mathcal{K} := \mathcal{K}(G^{(i)})$ is independent of i by structural equivalence since all models $G^{(i)}$ share the same system of differential equations. The same is also true of $\mathcal{Q}^{\max}(G^{(i)})$. The limit statement of Equation 6.4 is then immediate from Equation 6.3.

We may rewrite Equation D.2 in the proof of Theorem 6.2.1 as follows to accommodate non-deterministic initial probability distributions:

$$\begin{aligned}
\|\bar{\mathbf{N}}^{(i)}(t) - \bar{\mathbf{v}}_{\bar{\mathbf{n}}}(t)\|_\infty &= \left\|\bar{\mathbf{M}}^{(i)}(t) + \int_0^t \mathbf{f}(\bar{\mathbf{N}}^{(i)}(s))\,\mathrm{d}s - \int_0^t \mathbf{f}(\bar{\mathbf{v}}_{\bar{\mathbf{n}}}(s))\,\mathrm{d}s + \bar{\mathbf{N}}^{(i)}(0) - \bar{\mathbf{n}}\right\|_\infty \\
&\leq \|\bar{\mathbf{M}}^{(i)}(t)\|_\infty + \int_0^t \|\mathbf{f}(\bar{\mathbf{N}}^{(i)}(s)) - \mathbf{f}(\bar{\mathbf{v}}_{\bar{\mathbf{n}}}(s)\|_\infty\,\mathrm{d}s + \|\bar{\mathbf{N}}^{(i)}(0) - \bar{\mathbf{n}}\|_\infty \\
&\leq \|\bar{\mathbf{M}}^{(i)}(t)\|_\infty + \mathcal{K}\int_0^t \|\bar{\mathbf{N}}^{(i)}(s) - \bar{\mathbf{v}}_{\bar{\mathbf{n}}}(s)\|_\infty\,\mathrm{d}s + \|\bar{\mathbf{N}}^{(i)}(0) - \bar{\mathbf{n}}\|_\infty
\end{aligned}$$

We can then modify the event A in the proof of Theorem 6.2.2 as follows:

$$A := \left\{\sup_{t\in[0,T]} \|\bar{\mathbf{M}}^{(i)}_{\bar{\mathbf{n}}}(t)\|_\infty > \delta/2\right\} \cup \{\|\bar{\mathbf{N}}^{(i)}(0) - \bar{\mathbf{n}}\|_\infty > \delta/2\}$$

where $\delta := \epsilon\exp(-\mathcal{K}T)$ as before. Then following the method of that proof we may bound $\mathbb{P}(A)$ and thus $\mathbb{P}\left\{\sup_{t\in[0,T]} \|\bar{\mathbf{N}}^{(i)}(t) - \bar{\mathbf{v}}_{\bar{\mathbf{n}}}(t)\|_\infty > \epsilon\right\}$ as follows:

$$\mathbb{P}(A) \leq 2\mathcal{N}\exp(-D\mathcal{S}(G^{(i)})) + \mathbb{P}(\{\|\bar{\mathbf{N}}^{(i)}(0) - \bar{\mathbf{n}}\|_\infty > \delta/2\})$$

[6] That m is convex can be seen by noting that it has a non-negative second derivative everywhere.

where $D > 0$ is independent of i. Thus the limiting result for non-deterministic initial conditions also holds when $i \to \infty$.

To see the limit statement of Equation 6.5, we may re-define the stochastic process $\bar{\mathbf{M}}_{\bar{\mathbf{n}}}(t)$ in terms of the action-counting process $\bar{\mathbf{A}}^{(i)}_{\bar{\mathbf{n}}}(t)$:

$$\bar{\mathbf{M}}^{(i)}_{\bar{\mathbf{n}}}(t) := \bar{\mathbf{A}}^{(i)}_{\bar{\mathbf{n}}}(t) - \left(\int_0^t \mathbf{g}(\bar{\mathbf{N}}^{(i)}_{\bar{\mathbf{n}}}(s))\, \mathrm{d}s \right)$$

This can be seen to be a martingale in a similar manner to that of Appendix D.1.1 and the proof methodology of Appendix D.1.2 can then be followed to obtain the required result. □

D.1.4 Proof of Theorem 6.3.1

Proof. We begin with a technical lemma.

Lemma D.1.1. *For all* $\mathbf{x}_1$, $\mathbf{x}_2 \in \mathbb{R}^m$, *we have:*

$$(\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{P}(\hat{\mathbf{f}}(\mathbf{x}_1) - \hat{\mathbf{f}}(\mathbf{x}_2)) < -\alpha(\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{P}(\mathbf{x}_1 - \mathbf{x}_2)$$

for any matrix $\mathbf{P}$ *satisfying the hypotheses of Theorem 6.3.1.*

Proof. First consider the case where both $\mathbf{x}_1$, $\mathbf{x}_2 \in \Gamma_i$ for some i. Then:

$$\begin{aligned}
(\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{P}(\hat{\mathbf{f}}(\mathbf{x}_1) - \hat{\mathbf{f}}(x_2)) &= (\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{P}(\mathbf{A}_i \mathbf{x}_1 - \mathbf{A}_i \mathbf{x}_2) \\
&= (\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{P} \mathbf{A}_i (\mathbf{x}_1 - \mathbf{x}_2) \\
&= \frac{1}{2}(\mathbf{x}_1 - \mathbf{x}_2)^T (\mathbf{P}\mathbf{A}_i + \mathbf{A}_i^T \mathbf{P})(\mathbf{x}_1 - \mathbf{x}_2) \quad \text{since } \mathbf{P} \text{ is symmetric} \\
&< -\alpha(\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{P}(\mathbf{x}_1 - \mathbf{x}_2) \quad \text{by Equation 6.10}
\end{aligned}$$

Next we consider the general case of arbitrary $\mathbf{x}_1$, $\mathbf{x}_2 \in \mathbb{R}^m$. Consider the line segment joining these two points in $\mathbb{R}^m$. We write $\mathbf{z}_1 := \mathbf{x}_1$, $\mathbf{z}_k := \mathbf{x}_2$ and $\mathbf{z}_i$, $i = 2, \ldots, k-1$ for the intersection points of the line segment with distinct affine hyperplanes separating the regions of affine dynamics, not including the intersections with any such hyperplanes which contain the line entirely. We index the $\mathbf{z}_i$ in the order of intersection. In this manner, any consecutive pair of points $\mathbf{z}_i$ and $\mathbf{z}_{i+1}$ belong to some common region, say Γ_{j_i}. Then write $\mathbf{d} := \frac{\mathbf{x}_1 - \mathbf{x}_2}{\|\mathbf{x}_1 - \mathbf{x}_2\|_{\mathbf{P}}}$ where the norm $\|\mathbf{x}\|_{\mathbf{P}} := \sqrt{\mathbf{x}^T \mathbf{P} \mathbf{x}}$ for $\mathbf{x} \in \mathbb{R}^m$. Furthermore, we have $\mathbf{d} = \frac{\mathbf{z}_i - \mathbf{z}_{i+1}}{\|\mathbf{z}_i - \mathbf{z}_{i+1}\|_{\mathbf{P}}}$ for all $i = 1, \ldots, k-1$ since the $\mathbf{z}_i$ are on the same line segment and given in order.

We may then obtain:

$$\begin{aligned}
(\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{P}(\hat{\mathbf{f}}(\mathbf{x}_1) - \hat{\mathbf{f}}(\mathbf{x}_2)) &= \|\mathbf{x}_1 - \mathbf{x}_2\|_{\mathbf{P}} \mathbf{d}^T \mathbf{P}(\hat{\mathbf{f}}(\mathbf{x}_1) - \hat{\mathbf{f}}(\mathbf{x}_2)) \\
&= \|\mathbf{x}_1 - \mathbf{x}_2\|_{\mathbf{P}} \mathbf{d}^T \sum_{i=1}^{k-1} \mathbf{P}(\hat{\mathbf{f}}(\mathbf{z}_i) - \hat{\mathbf{f}}(\mathbf{z}_{i+1})) \\
&= \|\mathbf{x}_1 - \mathbf{x}_2\|_{\mathbf{P}} \sum_{i=1}^{k-1} \frac{(\mathbf{z}_i - \mathbf{z}_{i+1})^T \mathbf{P}(\hat{\mathbf{f}}(\mathbf{z}_i) - \hat{\mathbf{f}}(\mathbf{z}_{i+1}))}{\|\mathbf{z}_i - \mathbf{z}_{i+1}\|_{\mathbf{P}}}
\end{aligned}$$

Then since $\mathbf{z}_i$ and $\mathbf{z}_{i+1}$ both belong to the cell Γ_{j_i}, we have:

$$(\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{P}(\hat{\mathbf{f}}(\mathbf{x}_1) - \hat{\mathbf{f}}(\mathbf{x}_2)) < -\alpha \|\mathbf{x}_1 - \mathbf{x}_2\|_{\mathbf{P}} \sum_{i=1}^{k-1} \frac{(\mathbf{z}_i - \mathbf{z}_{i+1})^T \mathbf{P}(\mathbf{z}_i - \mathbf{z}_{i+1})}{\|\mathbf{z}_i - \mathbf{z}_{i+1}\|_{\mathbf{P}}}$$

So then:

$$\begin{aligned}(\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{P}(\hat{\mathbf{f}}(\mathbf{x}_1) - \hat{\mathbf{f}}(\mathbf{x}_2)) &< -\alpha \|\mathbf{x}_1 - \mathbf{x}_2\|_{\mathbf{P}} \sum_{i=1}^{k-1} \mathbf{d}^T \mathbf{P}(\mathbf{z}_i - \mathbf{z}_{i+1}) \\ &= -\alpha (\mathbf{x}_1 - \mathbf{x}_2)^T \sum_{i=1}^{k-1} \mathbf{P}(\mathbf{z}_i - \mathbf{z}_{i+1}) \\ &= -\alpha (\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{P}(\mathbf{x}_1 - \mathbf{x}_2)\end{aligned}$$

as required. □

Now we return to the proof of Theorem 6.3.1. Define the quadratic form $V : \mathbb{R}^m \to \mathbb{R}_+$ by $V(\mathbf{x}) := \frac{1}{2}(\mathbf{x} - \hat{\mathbf{v}}^*)^T \mathbf{P}(\mathbf{x} - \hat{\mathbf{v}}^*)$. Then $V(\hat{\mathbf{v}}^*) = 0$ and $V(\mathbf{x}) > 0$ for all $\mathbf{x} \neq \hat{\mathbf{v}}^*$ since $\mathbf{P}$ is positive definite.

Now by the chain rule we have $\frac{dV}{dt}(t) = \frac{\partial V}{\partial \mathbf{x}}(\hat{\mathbf{v}}(t)) \cdot \frac{\partial \hat{\mathbf{v}}}{\partial t}(t)$. Then since $\frac{\partial V}{\partial \mathbf{x}} = (\mathbf{x} - \hat{\mathbf{v}}^*)^T \mathbf{P}$ and $\frac{\partial \hat{\mathbf{v}}}{\partial t} = \hat{\mathbf{f}}(\hat{\mathbf{v}}(t))$, we have:

$$\begin{aligned}\frac{dV}{dt}(t) &= (\hat{\mathbf{v}}(t) - \hat{\mathbf{v}}^*)^T \mathbf{P} \hat{\mathbf{f}}(\hat{\mathbf{v}}(t)) \\ &= (\hat{\mathbf{v}}(t) - \hat{\mathbf{v}}^*)^T \mathbf{P}(\hat{\mathbf{f}}(\hat{\mathbf{v}}(t)) - \hat{\mathbf{f}}(\hat{\mathbf{v}}^*)) && \text{since } \hat{\mathbf{f}}(\hat{\mathbf{v}}^*) = 0 \\ &< -\alpha (\hat{\mathbf{v}}(t) - \hat{\mathbf{v}}^*)^T \mathbf{P}(\hat{\mathbf{v}}(t) - \hat{\mathbf{v}}^*) && \text{by Lemma D.1.1} \\ &< 0\end{aligned}$$

wherever $\hat{\mathbf{v}}(t) \neq \hat{\mathbf{v}}^*$ since P is positive definite.

Finally if $\|\mathbf{x}\| \to \infty$ then $\|\mathbf{x}\|_{\mathbf{P}} \to \infty$ by the equivalence of norms on $\mathbb{R}^m$ and then $V(\mathbf{x}) \to \infty$ so V is radially unbounded.

It thus follows that V fulfills all of the requirements to be a Lyapunov function and so verifies the global asymptotic stability of $\hat{\mathbf{v}}^*$ as required. □