Institutions
Markets
Technologies
IMT
INSTITUTE
FOR ADVANCED
STUDIES
LUCCA

#08
2013

**IMT LUCCA CSA TECHNICAL
REPORT SERIES 08
June 2013**

RA **Computer Science and Applications**

Applying Mean-field Approximation to Continuous Time Markov Chains

Anna Kolesnichenko
Alireza Pourranjabar
Valerio Senni

IMT LUCCA CSA TECHNICAL REPORT SERIES #08/2013

Piazza San Ponziano 6, 55100 Lucca

Research Area
Computer Science and Applications

Applying Mean-field Approximation to Continuous Time Markov Chains

Anna Kolesnichenko
Department of Design and Analysis of Communication Systems, University of Twente

Alireza Pourranjabar
Laboratory for Foundations of Computer Science, School of Informatics, University of Edinburgh

Valerio Senni
IMT Institute for Advanced Studies Lucca

Applying Mean-field Approximation to Continuous Time Markov Chains

Anna Kolesnichenko[1], Alireza Pourranjabar[2], and Valerio Senni[3]

[1] DACS, University of Twente, The Netherlands
kolesnichenkoav@ewi.utwente.nl
[2] LFCS, University of Edinburgh, UK
a.pourranjbar@sms.ed.ac.uk
[3] IMT Institute for Advanced Studies, Lucca, Italy
valerio.senni@imtlucca.it

Abstract. The mean-field analysis technique is used to perform analysis of a systems with a large number of components to determine the emergent deterministic behaviour and how this behaviour modifies when its parameters are perturbed. The computer science performance modelling and analysis community has found the mean-field method useful for modelling large-scale computer and communication networks. Applying mean-field analysis from the computer science perspective requires the following major steps: (1) describing how the agents populations evolve by means of a system of differential equations, (2) finding the emergent deterministic behaviour of the system by solving such differential equations, and (3) analysing properties of this behaviour either by relying on simulation or by using logics. Depending on the system under analysis, performing these steps may become challenging. Often, modifications of the general idea are needed. In this tutorial we consider illustrating examples to discuss how the mean-field method is used in different application areas. Starting from the application of the classical technique, moving to cases where additional steps have to be used, such as systems with local communication. Finally we illustrate the application of the simulation and fluid model checking analysis techniques.

1 Introduction

Mean Field Approximation originated in statistical physics [1] and it allows to find an estimate of the mean of a hard to compute distribution. This technique is useful to study the behavior of stochastic processes with a very large state space (e.g. in the study of systems with a large number of particles), where Monte Carlo simulations are impractical. In those systems, a first approximation of the behavior is obtained by replacing the effect of the other particles over a given particle by a single averaged effect and studying this two-body problem [29,37]. Beyond physics, this approximation technique finds applications in studies of epidemics models [30], queueing theory [6,1], and network performance [36,15].

The stochastic systems we are interested in this tutorial typically consist of a relatively small number of particle types replicated many times to form large

populations. Mean-field approximation is used to model and analyze efficiently the emergent behavior of such large-scale systems. Classical applications of this technique are generally based on two abstractions. The first one ignores agents identities and, rather than looking at the individual agent behavior, observes the system at the level of populations [28]. The second abstraction ignores the spatial distribution of the agents across the system locations, and the particles are assumed to be uniformly spread across the system space (in chemistry this idea is embodied in the notion of well-stirred chemical reaction [22,43]). In this tutorial we illustrate both a classical application in full details (Section 4) and a more sophisticated modeling of space that consider the effect agents locations have on the emergent behavior of the system (Section 5).

The core idea of mean-field approximation is to approximate the *mean* dynamics of a Markov population process through a system of differential equations [33]. This is a reliable approximation if the considered system shows an emergent behavior (e.g. by showing convergence to zero of variance) and when the population size is sufficiently large. Under those conditions the random dynamics of the Markov process are very close to the deterministic dynamics defined through the differential equations. A further interesting property is that the joint probabilities of assuming a certain state configuration become disjoint and, thus, one can focus on one particular individual rather than on the population dynamics, given in terms of the solution of a differential equation. This gives enormous benefits in terms of cost of the analysis.

A closely related approximation technique is known as *moment closure* [21]. This technique allows to estimate the moments of a stochastic process by truncating the moment equations. This results in a closed system of equations whose solution can be attempted. Mean-field approximation can be seen as a form of moment closure where the second moment (variance), as well as the higher moments, have been truncated (i.e. set to zero). The first-order approximation is often very coarse and can lead to misleading results [39]. In practice, however, it can be used to gain some insights about the average, global behavior of the system at a relatively low cost. Then, further study of the system is required, for example considering approximations of higher moments.

When first-order or mean-field approximation is applied, the resulting model can be described in terms of a deterministic system, as mentioned previously. This is often referred to, in the literature, as *deterministic approximation* [4,11].

Another related technique is called *linear noise approximation*, which is frequently used to find approximate solutions of the Chemical Master Equation by giving an estimate of the second moment of this equation [43].

Depending on the type of Markov process one considers, as well as on how the model scales with increasing population, one needs to rely on different mean field results. In particular, if we consider Discrete Time Markov Chains (DTMC), we can have either mean field limits in discrete time (where all individuals try to perform a transition at each step, thus assuming a *synchronous* semantics) or in continuous time (where a few individuals try to perform a transition, thus assuming an *asynchronous* semantics). If we consider Continuous Time Markov

Chains (CTMC) we have only limits in continuous time. The first result on the deterministic approximation of a sequence of CTMC models can be found in Kurtz [33]. For the case of DTMC models (which we do not treat in this tutorial) one can refer to [3,36]. On the basis of the limit one obtains, the approximating dynamical system will be expressed either in terms of finite-difference equations, for discrete time, or ordinary differential equations, for continuous time.

Continuous Time Markov Chains are often used to provide a stochastic semantics to process algebras used in performance modelling of computer systems [26]. However, stochastic process algebra models of realistic size can easily result in underlying state spaces of intractable size. In that context a technique called *fluid-flow approximation* [27] has been used to construct a continuous state-space representation of the underlying discrete state-space, and ordinary differential equations are used to describe the dynamics of these systems. This technique is justified by results on mean-field approximation of Continuous Time Markov Chains [42,28,25]. Indeed, the notion of fluid approximation has been used in various contexts such as Petri Nets, and relies on the idea that a discrete variable can be approximated using a continuous variable [40]. In the context of mean-field approximation of Continuous Time Markov Chains, the fluidification is essentially involved when discrete stochastic variables counting the populations are replaced by continuous variables.

In our tutorial we focus on CTMC models and their continuous-time approximation using ordinary differential equations. The goal of this paper is to provide an example-guided tutorial to the application of fluid approximation, including fluid model checking [8]. The interested reader can find very complete and detailed tutorials in [10,11], treating both Continuous Time Markov Chains and Discrete Time Markov Chains. A more technical survey of the topic and related mathematical results can be found in [18].

2 Preliminaries

In the attempt of providing a self contained introductory tutorial to mean field approximation of Continuous Time Markov Chains and in order to allow the reader to follow the details of the examples we present, we briefly recall in this section the principal mathematical notions used in this tutorial.

Let us consider a countable domain $\mathcal{D}$ (we assume $\mathcal{D} \subset \mathbb{R}^n$). A discrete random variable is a distribution over the discrete domain $\mathcal{D}$. For a thorough treatment of the theory of probability the reader can refer to [5]. We follow the notation of [11].

A CTMC is a (dense) time-indexed family of discrete random variables (i.e. distributions) over a countable state space. It can be seen as a description of the evolution in (continuous) time of a discrete random variable.

Definition 1 (Continuous Time Markov Chain). *A $\mathcal{D}$-valued homogeneous Continuous Time Markov Chain (CTMC) $\mathbf{X}(t)$ is an $\mathbb{R}_{\geq 0}$-indexed family $\{\mathbf{X}_t \mid t \in \mathbb{R}_{\geq 0}\}$ of $\mathcal{D}$-valued Discrete Random Variables such that:*

1. $\mathbb{P}\{\mathbf{X}(t_k)=\mathbf{d}_k \mid \mathbf{X}(t_0)=\mathbf{d}_0, \ldots, \mathbf{X}(t_h)=\mathbf{d}_h\} = \mathbb{P}\{\mathbf{X}(t_k)=\mathbf{d}_k \mid \mathbf{X}(t_h)=\mathbf{d}_h\}$,
   for $t_0 < \ldots < t_h < t_k \in \mathbb{R}$ *and* $d_0, \ldots, d_h, d_k \in \mathcal{D}$, *and* *(memoryless)*

2. $\mathbb{P}\{\mathbf{X}(t+\delta)=\mathbf{d}_1 \mid \mathbf{X}(t)=\mathbf{d}_1\} = \mathbb{P}\{\mathbf{X}(u+\delta)=\mathbf{d}_2 \mid \mathbf{X}(u) = \mathbf{d}_2\}$,
   for $t, u, \delta \in \mathbb{R}$ *and* $d_1, d_2 \in \mathcal{D}$. *(time homogeneity)*

For a CTMC, we can define the initial probability distribution $\pi(0) : \mathcal{D} \to [0,1]$ and the probabilistic transition matrix $\mathbf{P} : \mathcal{D}^2 \to [0,1]$, which, relying on properties (1) and (2) above, can be defined as $\mathbf{P}(\mathbf{d}_1, \mathbf{d}_2) = \mathbb{P}\{\mathbf{X}_\delta = \mathbf{d}_2 \mid \mathbf{X}_0 = \mathbf{d}_1\}$, for any $\mathbf{d}_1, \mathbf{d}_2 \in \mathcal{D}$ and $\delta > 0 \in \mathbb{R}$. This transition probability depends on δ, that represents the time spent at $\mathbf{d}_1$ before the transition takes place. With each state $\mathbf{d}$ we can associate a continuous random variable $\mathbf{X}_\mathbf{d}$ representing the time spend in $\mathbf{d}$ before any outgoing transition occurs, called the *sojourn time*. One can show that memoryless of Markov chains entails that sojourn time is an exponentially distributed continuous random variable, with a given rate $\lambda_\mathbf{d}$. Let $\Lambda : \mathcal{D} \to \mathbb{R}_{>0}$ be the exit rate function, we can define the *infinitesimal generator matrix* $\mathbf{Q} : \mathcal{D}^2 \to \mathbb{R}$ as $\mathbf{Q}(\mathbf{d}_1, \mathbf{d}_2) = \Lambda(\mathbf{d}_1) \cdot \mathbf{P}(\mathbf{d}_1, \mathbf{d}_2)$, for $\mathbf{d}_1 \neq \mathbf{d}_2 \in \mathcal{D}$, and $\mathbf{Q}(\mathbf{d}, \mathbf{d}) = -\sum_{\mathbf{d}' \neq \mathbf{d}} \mathbf{Q}(\mathbf{d}, \mathbf{d}')$.

As a consequence of these observations, a given CTMC can be equivalently represented either as the tuple $\langle \mathcal{D}, \mathbf{P}, \Lambda, \pi(0) \rangle$ or as the tuple $\langle \mathcal{D}, \mathbf{Q}, \pi(0) \rangle$ [35]. In the rest of the paper, depending on the context, we rely on both representations of CTMCs. A CTMC can be *labelled*, that is, it can include a state-labelling function $L : \mathcal{D} \to 2^{AP}$ assigning to each state a set of atomic properties in AP.

In this paper we consider population models, that are Markov chains modelling the evolution of the number of individuals living within a fixed number of classes. These models are used in biology and chemistry, as well as in telecommunications and queueing theory [3,16,28,41]. Population models are also adopted as abstractions of large Markov models, e.g. obtained by parallel composition of several CTMC models. Such large models are unmanageable for the purpose of analysis, due to known problems of state space explosion, and are not suitable for direct application of classic analysis techniques such as simulation and model checking.

Population models are obtained from the original models through two abstraction steps [42]. The first abstraction consists in identifying a number of classes or macro-states and representing the *number of processes* in a given class rather than the state of each process, thereby loosing the identity of the single process. The second abstraction consists in considering the so-called occupancy measure, that is the *fraction of the population* rather than the actual amount of individuals. This second abstraction can also be thought of as a normalization step, that allows us to compare population models with different initial populations.

Example 1. Consider a system with N agents, where $\mathbf{S}_i^{(N)}(t) \in \{1, \ldots, n\}$ denotes the state of agent i at time t. The first abstraction discussed above consists in considering the quantity $\mathbf{X}_i^{(N)}(t) = \sum_{j=1}^{N} \mathbf{1}\{\mathbf{S}_j^{(N)}(t) = i\}$, which denotes

the number of processes in state i at time t ($\mathbf{1}\{\varphi\}$ is the function equal to 1 when the property φ holds, known as *indicator function*). The second abstraction consists in considering the fraction $\overline{\mathbf{X}}_i^{(N)}(t) = \frac{1}{N}\mathbf{X}_i^{(N)}(t)$ of the processes in state i. As a consequence of these abstractions, while the size of the state $\mathbf{S}^{(N)}(t) = \langle \mathbf{S}_1^{(N)}(t), \ldots, \mathbf{S}_N^{(N)}(t)\rangle$ of the system depends on the population, the size of the state $\langle \mathbf{X}_1^{(N)}, \ldots, \mathbf{X}_n^{(N)}\rangle$ of the model is independent of the population. On the contrary, while the state space of $\mathbf{S}_i^{(N)}(t)$ ranges on the fixed set $\{1, \ldots, n\}$, the state of the abstraction ranges over the set $\{0, \frac{1}{N}, \frac{2}{N}, \ldots, 1\}^n$, which in the limit becomes the continuous interval $[0,1]^n \subseteq \mathbb{R}^n$.

In the following we assume this abstraction has already been done and we discuss directly of quantities $\mathbf{X}^{(N)}$ and $\overline{\mathbf{X}}^{(N)}$. Sections 4 and 5, dealing with concrete systems and their models, provide more details and examples concerning these two abstraction steps. Let us now formalize these notions and discuss some global measures of population models that allow us to analyse emergent behaviours of these models.

Definition 2 (Population Continuous Time Markov Chain Model). *A Population Continuous Time Markov Chain (PCTMC) model $\mathcal{X}$ is a tuple $\langle \mathbf{X}, \mathcal{D}, \mathcal{T}, \mathbf{d}_0\rangle$ such that:*

1. $\mathbf{X} = (X_1, \ldots, X_n)$ *is a vector of* variables*, taking values in a countable domain $\mathcal{D}_i \subset \mathbb{R}$,*
2. $\mathcal{D} = \prod_i \mathcal{D}_i$ *is the state space of the model,*
3. $\mathcal{T} = \{\tau_1, \ldots, \tau_m\}$ *is a set of transitions such that $\tau_i = \langle \ell, \mathbf{v}, r\rangle$ and*

   ℓ is the transition label,
   $\mathbf{v} \in \mathbb{R}^n$ is the state-change vector,
   $r : \mathcal{D} \to \mathbb{R}_{\geq 0}$ is the transition rate function, such that $r(\mathbf{d}) = 0$ if $\mathbf{d}+\mathbf{v} \notin \mathcal{D}$;
4. $\mathbf{d}_0 \in \mathcal{D}$ *is the initial state of the model.*

Let us describe this model. $\mathbf{X}_i(t)$ indicates the number of individuals residing in state $i \in \{1, 2, \ldots, n\}$ at time t. The system population at time t is $N(t) = \sum_{i=1}^n X_i(t)$, the initial population is $\mathbf{X}(0) = \mathbf{d}_0$. The execution of a transition τ consists in performing an action with label ℓ which modifies the current population $\mathbf{d}$ into the new population $\mathbf{d}'$, where $\mathbf{d}' - \mathbf{d} = \mathbf{v}_\tau$ and $\mathbf{v}_\tau$ is the corresponding state-change vector. No assumptions on balance in these transitions is taken since, in general, we allow the modelling of birth/death processes and the population need not be preserved. The population can also be modified in terms of *fractions* of individuals, but the condition on the transition rate function ensures the reachable states belong to the fixed, countable state space.

A PCTMC model $\mathcal{X} = \langle \mathbf{X}, \mathcal{D}, \mathcal{T}, \mathbf{d}_0\rangle$ has an underlying CTMC process $\mathbf{X}(t) = \langle \mathcal{D}, \mathbf{P}, \Lambda, \pi(0), L\rangle$, where $\mathbf{P}$ and Λ are obtained by computing the infinitesimal generator matrix as described below. The initial probability distribution $\pi(0) : \mathcal{D} \to [0,1]$ is such that $\pi(0)(\mathbf{d}_0) = 1$ and $\pi(0)(\mathbf{d}) = 0$ for any $\mathbf{d} \neq \mathbf{d}_0$.

The transition rate from $\mathbf{d}$ to $\mathbf{d}'$ is the sum of the rates of the outgoing transitions from $\mathbf{d}$ whose state-change vector leads to $\mathbf{d}'$: $\mathbf{Q}(\mathbf{d},\mathbf{d}') = \sum_{\{\tau\in\mathcal{T}\,|\,\mathbf{d}'=\mathbf{d}+\mathbf{v}_\tau\}} r_\tau(\mathbf{d})$, if $\mathbf{d}' \neq \mathbf{d}$, and $\mathbf{Q}(\mathbf{d},\mathbf{d}) = -\sum_{\mathbf{d}'\neq\mathbf{d}} \mathbf{Q}(\mathbf{d},\mathbf{d}')$ otherwise. We also assume to have a state-labelling function $L : \mathcal{D} \to 2^{AP}$.

When studying systems with a large number of components, we consider a sequence $(\mathcal{X}^{(i)})_I = \mathcal{X}^{(i_0)}\mathcal{X}^{(i_1)}\ldots$ of PCTMC models, indexed over a set $I \subseteq \mathbb{N}$. The notation $\mathcal{X}^{(i)} = \langle \mathbf{X}^{(i)}, \mathcal{D}^{(i)}, \mathcal{T}^{(i)}, \mathbf{d}_0^{(i)}\rangle$ indicates that all the components of a PCTMC model depend on the parameter i, for $i \in I$. To each model $\mathcal{X}^{(i)}$ we associate a size γ_i, provided by a function $\gamma : I \to \mathbb{R}_{\geq 0}$. In most cases the sequence of PCTMC models is indexed over the entire $\mathbb{N}$ and the size is exactly the population, that is the total number of components/agents in the system. We indicate this choice by $(\mathcal{X}^{(i)})_{\mathbb{N}}$, and fixing the size γ_i to be the population N. However, in general the population may depend on time (such as in the birth/death processes), thus not being a constant of the model.

We now introduce some notions to describe the *global* behavior of a PCTMC model $\mathcal{X}$. The *exit rate* $R_\mathcal{X} : \mathcal{D} \to \mathbb{R}$ describes the rate of the event that an outgoing transition happens from a given state.

$$R_\mathcal{X}(\mathbf{d}) = \sum_{\tau\in\mathcal{T}} r_\tau(\mathbf{d})$$

In [3] this notion is called *intensity*.

The *mean increment* $\mu_\mathcal{X} : \mathcal{D} \to \mathbb{R}^n$ describes the average variation of each variable in a discrete PCTMC step, and it is defined as the sum of the variations induced by each transition, multiplied by the probability for that transition to happen.

$$\mu_\mathcal{X}(\mathbf{d}) = \sum_{\tau\in\mathcal{T}} \mathbf{v}_\tau \frac{r_\tau(\mathbf{d})}{R(\mathbf{d})}$$

where we assume $R(\mathbf{d}) > 0$. Finally, we consider the *mean dynamics* (also called *drift*) $F_\mathcal{X} : \mathcal{D} \to \mathbb{R}^n$ that describes the average local variation of the PCTMC with respect to the time elapse.

$$F_\mathcal{X}(\mathbf{d}) = R_\mathcal{X}(\mathbf{d})\mu_\mathcal{X}(\mathbf{d}) = \sum_{\tau\in\mathcal{T}} \mathbf{v}_\tau r_\tau(\mathbf{d})$$

Any model $\mathcal{X}^{(i)}$ of a sequence $(\mathcal{X}^{(i)})_I$ has his own parameters $R_{\mathcal{X}^{(i)}}$, $\mu_{\mathcal{X}^{(i)}}$, $F_{\mathcal{X}^{(i)}}$. In the mean-field approximation theorem we are interested into parameters $R_{(\mathcal{X}^{(i)})_I}$, $\mu_{(\mathcal{X}^{(i)})_I}$, $F_{(\mathcal{X}^{(i)})_I}$ characterizing a sequence of PCTMC models. Indeed, if such parameters exist and satisfy certain scaling assumptions, we are able to characterize the limit behaviour of the sequence $(\mathcal{X}^{(i)})_I$ in terms of those parameters. In the following section we provide sufficient conditions under which those parameters can be found and a theorem that allows us to define the dynamics of the sequence $(\mathcal{X}^{(i)})_I$ using those parameters.

Indeed, in order to be able to compare models of different size, we need to transform each model $\mathcal{X}^{(i)} = \langle \mathbf{X}^{(i)}, \mathcal{D}^{(i)}, \mathcal{T}^{(i)}, \mathbf{d}_0^{(i)}\rangle$ of a sequence $(\mathcal{X}^{(i)})_I$ into

the corresponding, normalized model $\overline{\mathcal{X}}^{(i)} = \langle \overline{\mathbf{X}}^{(i)}, \overline{\mathcal{D}}^{(i)}, \overline{\mathcal{T}}^{(i)}, \overline{\mathbf{d}}_0^{(i)} \rangle$, obtained by applying a *normalization* operator $\bar{\cdot}$, defined as follows:

1. $\overline{\mathbf{X}}^{(i)}$ is the new vector of state variables,
2. $\overline{\mathcal{D}}^{(i)} = \{\overline{\mathbf{d}} \mid \mathbf{d} \in \mathcal{D}^{(i)}\}$, where $\overline{\mathbf{d}} = \frac{1}{\gamma_i}\,\mathbf{d}$, for every $\mathbf{d} \in \mathcal{D}$,
3. $\overline{\mathcal{T}} = \{\overline{\tau} \mid \tau \in \mathcal{T}^{(i)}\}$,
   where for a transition $\tau = \langle \ell, \mathbf{v}^{(i)}, r^{(i)} \rangle$ the normalized transition is $\overline{\tau} = \langle \ell, \overline{\mathbf{v}}^{(i)}, \overline{r}^{(i)} \rangle$, with $\overline{\mathbf{v}}^{(i)} = \frac{1}{\gamma_i}\mathbf{v}^{(i)}$ and $\overline{r}^{(i)}(\overline{\mathbf{d}}) = r^{(i)}(\gamma_i\,\overline{\mathbf{d}})$, for every $\overline{\mathbf{d}} \in \overline{\mathcal{D}}$.

As an effect of normalization, we have the relation $\overline{\mathbf{X}}^{(i)} = \frac{1}{\gamma_i}\,\mathbf{X}^{(i)}$ between the state-space of the normalized model and that of the non-normalized one. The normalized state space $\overline{\mathbf{X}}^{(i)}$ is also known in the literature as *occupancy measure*. As a consequence of normalization, any model $\overline{\mathcal{X}}^{(i)}$ of a sequence $(\mathcal{X}^{(i)})_I$ has his own parameters $R_{\overline{\mathcal{X}}^{(i)}}$, $\mu_{\overline{\mathcal{X}}^{(i)}}$, $F_{\overline{\mathcal{X}}^{(i)}}$.

3 Mean-field Approximation

The core idea of the mean-field approximation is that, under certain assumption on the dynamics of the population and when the size of the PCTMCs grows (i.e. in the limit), the drift vectors become coherent. In particular, the *variance* of the system becomes zero so the approximation over the average behaviour is faithful. Therefore, the average behaviour can be modelled considering the *unique solution* of a system of Ordinary Differential Equations defined by using the limit mean dynamics (the drift) of the PCTMC family.

The ODE approximation of the sequence of CTMC models is defined on a continuous domain, while each model in the sequence has its state space on a countable domain. To re-conciliate these two domains, we consider a *closed* set $E \subset \mathbb{R}^n$ that contains the state space of each model in the sequence: $\bigcup_I \overline{\mathcal{D}}^{(i)} \subseteq E$.

An important requirement for the mean-field approximation theorem is *convergence of initial conditions*, which can be understood as the need for the all the PCTMC models of a sequence to have the same proportion of individuals among the various populations. The limit of these initial conditions constitute the initial condition for the ODE that approximates the mean dynamics.

Definition 3 (Convergence of Initial Conditions). *A sequence* $(\mathcal{X}^{(i)})_I$ *satisfies* convergence of initial conditions *if there is a point* $\mathbf{d}_0 \in E$ *such that, when considering the initial conditions of the normalized models,* $\lim_{i\to\infty} \overline{\mathbf{d}}_0^{(i)} = \overline{\mathbf{d}}_0$.

3.1 Density Dependence

As a first step, we consider a restricted version of the mean-field approximation theorem, applicable to the so-called *density dependent* sequences of models, defined as follows.

Definition 4 (Density Dependence). *The sequence* $(\mathcal{X}^{(i)})_I = \mathcal{X}^{(i_0)}\mathcal{X}^{(i_1)}\dots$ *of PCTMC models is density dependent if and only if:*

1. *the size grows linearly in* i*:* $\gamma_i \in \Theta(i)$*;*
2. *for any transition, the corresponding state-change vector is independent of the parameter of the sequence:*
   for any transition τ *there is a vector* $\mathbf{u}_\tau$ *such that, for any* $i \in I$*,* $\mathbf{v}_\tau^{(i)} = \mathbf{u}_\tau$*;*
3. *the rate functions depend on the parameter* i *only in terms of normalization:*
   for any transition τ *there is a function* $g_\tau : E \to \mathbb{R}$ *such that, for any* $i \in I$*,* $r_\tau^{(i)}(\mathbf{d}) = \gamma_i\, g_\tau(\frac{1}{\gamma_i}\,\mathbf{d})$*, for all* $\mathbf{d} \in \mathcal{D}^{(i)}$*.*

Density dependent sequences of PCTMC have rates and mean dynamics that scale together with the model size so that in the normalized models they are independent of the size. This allows to find easily the limit of the mean dynamics and to use it to define the field for the ODE that approximates the mean dynamics. These observations are formalized by the following properties.

A normalized model $\overline{\mathcal{X}}^{(i)}$ of a density dependent sequence has the following (global) properties:

1. for any state, the exit rate grows linearly with the model size:
   $$R_{\overline{\mathcal{X}}^{(i)}}(\overline{\mathbf{d}}) = \sum_{\tau \in \overline{\mathcal{T}}^{(i)}} r_\tau^{(i)}(\overline{\mathbf{d}}) = \sum_{\tau \in \overline{\mathcal{T}}^{(i)}} \gamma_i\, g_\tau(\overline{\mathbf{d}}) \qquad (\dagger)$$
   therefore, since g_τ does not depend on i and the size is linear in i, in the normalized domain $R_{\overline{\mathcal{X}}^{(i)}} \in \Theta(i)$;
2. the mean dynamics does not depend on i:
   $$F_{\overline{\mathcal{X}}^{(i)}}(\overline{\mathbf{d}}) = \sum_{\tau \in \overline{\mathcal{T}}^{(i)}} \overline{\mathbf{v}}_\tau^{(i)} r_\tau^{(i)}(\overline{\mathbf{d}}) = \sum_{\tau \in \mathcal{T}^{(i)}} \mathbf{u}_\tau g_\tau(\overline{\mathbf{d}}) \qquad (\ddagger)$$
   let us denote by $F_{(\mathcal{X}^{(i)})_I}$ the mean dynamics of the sequence $(\mathcal{X}^{(i)})_I$.

In [3], property (1) above corresponds to the notion of *vanishing intensity*. As a consequence of those properties, under density dependence, we are able to calculate a mean dynamics which is common to all the models of the sequence. The following step is now to evaluate the behaviour of each model of the sequence w.r.t. the limit mean dynamics. The mean-field approximation theorem that we are going to introduce states that the variance of the trajectories becomes small as the size of the model grows and converges to the limit mean dynamics.

Let us now fix some notation for the remaining part of this section. Assume $(\overline{\mathcal{X}}^{(i)})_I$ is a sequence of normalized population models, $\overline{\mathcal{X}}^{(i)}$ be one of these models, and $\overline{\mathbf{X}}^{(i)}(t)$ be the underlying Markov process. Finally, let $\overline{\mathbf{x}}(t)$ the solution of the initial value problem $\frac{d\overline{\mathbf{x}}(t)}{dt} = F(\overline{\mathbf{x}}(t))$ and $\overline{\mathbf{x}}(0) = \overline{\mathbf{d}}_0$, for a given (Lifschitz-continuous) field F.

We now state a first version of the mean-field approximation theorem, based on density dependence and assuming globally Lipshitz-continuous dynamics.

Furthermore, in Figure 1 we recapitulate how the main notions illustrated in this section are combined into a systematic approach to applying mean-field approximation to PCTMCs.

Theorem 1 (Mean-field Approx. of Density Dependent PCTMCs). *Let the sequence* $(\mathcal{X}^{(i)})_I = \mathcal{X}^{(i_0)}\mathcal{X}^{(i_1)}\ldots$ *of PCTMC models be density dependent and enjoy convergence of initial conditions to the point* $\overline{\mathbf{d}}_0 \in E$*. Let the drift* $F_{(\mathcal{X}^{(i)})_I}$ *be a Lipschitz-continuous vector field and* $\overline{\mathbf{x}}(t)$ *the solution of the initial value problem* $\overline{\mathbf{x}}(0) = \overline{\mathbf{d}}_0$ *and* $\frac{d\overline{\mathbf{x}}(t)}{dt} = F_{(\mathcal{X}^{(i)})_I}(\overline{\mathbf{x}}(t))$*. Then, for any finite time horizon* $T < \infty$

$$\mathbb{P}\{\lim_{i\to\infty}(\sup_{0\leq t\leq T}\|\overline{\mathbf{X}}^{(i)}(t) - \overline{\mathbf{x}}(t)\|) = 0\} = 1$$

The theorem states that the sequence $(\mathcal{X}^{(i)})_I$ of population models *converges almost surely* [5] to the dynamics of the ODE. That is, if we compare the behaviour of the underlying Markov process $\overline{\mathbf{X}}^{(i)}(t)$ with the solution $\overline{\mathbf{x}}(t)$ of the dynamical systems defined through the limit mean drift field, we observe that, as the model size grows, the worst mean square distance converges to zero almost surely for *any finite time horizon*. As a consequence, as the model size grows, the dynamics of the PCTMC becomes *deterministic* and can be faithfully approximated by the (possibly nonlinear) dynamics of $\overline{\mathbf{x}}(t)$.

We are now ready to describe a systematic approach to the application of the mean-field approximation, illustrated in Figure 1. The first step consists in definig a sequence $(\mathcal{X}^{(i)})_I$ of population models parameterized in their size as indicated in Def. 2. One can also rely on higher-level languages such as those based on process algebras. A notable example is PEPA, that has a stochastic, lumped semantics based on the idea of counting process types which is close to that of Def. 2, from which ODEs are derived [42].

The second step consists in choosing appropriate initial conditions, according to Def. 3. Then, it is necessary to check satisfaction of Def. 4. If all the requirements are satisfied, we can derive a limit drift matrix as indicated by (‡) which must be checked for Lipschitz continuity. The initial conditions together with the limit drift are then used to define the initial value problem of Theorem 1, which is ensured to be coherent to the dynamics of $(\mathcal{X}^{(i)})_I$ for large i.

In Sec. 4 we illustrate an application of this systematic approach on a concrete example modeling the spread of computer viruses.

3.2 Beyond Density Dependence

For models considered in practice, however, the assumption of density dependence may be too restrictive [18]. Furthermore, also the assumption of (global) Lipschitz continuity of $F_{(\mathcal{X}^{(i)})_I}$ can be unrealistic [7]. Therefore, we now consider a more general version of the mean-field approximation theorem, having less strict requirements and applied to *prefixes* of trajectories rather than to full model trajectories.

1. Define a sequence of (normalized) population models $(\overline{\mathcal{X}}^{(i)})_I$, in terms of a *parameterized* model $\overline{\mathcal{X}}^{(i)}$ defined following Def. 2;
2. Choose initial conditions $\overline{\mathbf{d}}_0$ satisfying Def. 3;
3. Check density dependence of $(\overline{\mathcal{X}}^{(i)})_I$ according to Def. 4;
4. Apply (‡) to compute the drift matrix $F_{\overline{\mathcal{X}}^{(i)}}$ and construct the system of Ordinary Differential Equations with initial conditions $\overline{\mathbf{d}}_0$;
5. Check Lipschitz-continuity of $F_{\overline{\mathcal{X}}^{(i)}}$;
6. Analyze the solution $\overline{\mathbf{x}}(t)$ of this initial value problem, which approximates the mean behavior of $\overline{\mathcal{X}}^{(i)}$ for large values of i as in Theorem 1.

Fig. 1: The general procedure for applying mean-field approximation.

We consider a set S which is *open relatively to* the set E and contains the state-space of the family of PCTMC models under consideration[4]. We formulate all the scaling assumptions w.r.t. dynamics of the family of PCTMC models that live within S. In particular, we consider the parametric space $S^{(i)} = \overline{\mathcal{D}}^{(i)} \cap S$.

The first requirement concerns the behaviour of the system mean dynamics (drift) when the size grows.

Definition 5 (Convergence of Drift). *A sequence $(\mathcal{X}^{(i)})_I$ of PCTMC models satisfies* convergence of drift *if there exists a Lipschitz vector field $F : E \to \mathbb{R}^n$ such that the mean dynamics $F_{(\overline{\mathcal{X}}^{(i)})_I}$ of the* normalized *sequence converge uniformly to F:*

$$\lim_{i\to\infty} \sup_{\overline{\mathbf{d}}\in S^{(i)}} \|F_{(\overline{\mathcal{X}}^{(i)})_I}(\overline{\mathbf{d}}) - F(\overline{\mathbf{d}})\| = 0$$

In this definition we require Lipschitz continuity of F and convergence only *within* $S^{(i)}$. If convergence of drift is satisfied, we can study, within $S^{(I)}$, the behaviour of the solution of the initial value problem $\frac{d\overline{\mathbf{x}}(t)}{dt} = F(\overline{\mathbf{x}}(t))$ with $\overline{\mathbf{x}}(0) = \overline{\mathbf{d}}_0$, rather than the original model $\overline{\mathcal{X}}^{(i)}$. However, we are unable to evaluate the error we commit in this approximation.

The second requirement concerns the effect on exit rates and jump magnitude of model size growth. In particular, we require that the variance of the system dynamics (which is considered to be noise w.r.t. to the deterministic dynamics) goes to zero.

[4] Recall that sets are defined to be open w.r.t. a topology: here we assume the topological space $\mathbb{R}^n$. If E is a subset of $\mathbb{R}^n$, then a set S is *open relatively* to E if $S = U \cap E$, for some open set U in $\mathbb{R}^n$. As a simple example, let S be the set $(0, 1) \subset \mathbb{Q}$ (the rational numbers). Now, if $E = \mathbb{Q}$ then S is open w.r.t. E, but if $E = \mathbb{R}$ then S is *not* open w.r.t. E (no open subset of $\mathbb{R}$, intersected with E, allows to define S).

Definition 6 (Convergence to Zero of Noise). *A sequence* $(\mathcal{X}^{(i)})_I$ *of PCTMC model satisfies* convergence to zero of noise *if, once normalized:*

(1) the exit rate is bounded*, for any size i:*
for any $i \in I$*, there is* $\Lambda_i \in \mathbb{R}$ *such that* $\Lambda_i < \infty$ *and* $\sup_{\overline{\mathbf{d}} \in S^{(i)}} R_{\overline{\mathcal{X}}^{(i)}}(\overline{\mathbf{d}}) = \Lambda_i$*;*

(2) the magnitude of jumps goes to zero*, as i increases:*
for any $i \in I$*, there is* $J_i \in \mathbb{R}$ *such that* $\max_{\tau \in \mathcal{T}^{(i)}} \|\mathbf{v}_\tau^{(i)}\| = J_i$ *and* $J_i \in O(i^{-1})$*;*

(3) *jump magnitude and exit rate satisfy* $J_i^2 \Lambda_i \in O(i^{-1})$.

The notions of convergence of drift and convergence to zero of noise depend and are limited to the restricted state space $S^{(i)}$. One can prove that density dependence implies convergence of drift and convergence to zero of noise.

Let us assume that we are given a relatively open subset S of the state space E, a vector field F Lifschitz in S, and an initial value $\overline{\mathbf{d}}_0 \in S$. The following, more general version of the mean-field approximation theorem holds for prefixes of the PCTMC behavior that live within S. In particular, it relies on a notion of *exit time* from the region S: let the exit time from S of the markov process $\overline{\mathbf{X}}^{(i)}(t)$ be defined as $\zeta^{(i)}(S) = \inf\{t \geq 0 \,|\, \overline{\mathbf{X}}(t) \notin S\}$ and the exit time from S of the ode solution $\overline{\mathbf{x}}(t)$ be defined as $\zeta(S) = \inf\{t \geq 0 \,|\, \overline{\mathbf{x}}(t) \notin S\}$.

Theorem 2 (Mean-field Approximation of PCTMCs). *Let the sequence* $(\mathcal{X}^{(i)})_I = \mathcal{X}^{(i_0)}\mathcal{X}^{(i_1)} \ldots$ *of PCTMC models and a given vector field F (Lifschitz in S) satisfy convergence of initial conditions, convergence of drift, and convergence to zero of noise. For any finite time horizon* $T < \zeta(S)$*:*

1. $\lim_{i \to \infty} \mathbb{P}\{\zeta^{(i)}(S) < T\} = 0$
2. *for all* $\varepsilon \in \mathbb{R}_{>0}$*,* $\lim_{i \to \infty} \mathbb{P}\{\sup_{0 \leq t \leq T} \|\overline{\mathbf{X}}^{(i)}(t) - \overline{\mathbf{x}}(t)\| > \varepsilon\} = 0$

This theorem states that, for any horizon within the exit time $\zeta(S)$, (i) when the size of the model grows, the probability the PCTMC model exits S before the exit time of the ode solution is zero, and (ii) the sequence $(\mathcal{X}^{(i)})_I$ of population models *converges in probability* [5] to the dynamics of the ode. That is, the probability of observing a difference bigger than ε between any point of a trajectory of the Markov process and the solution of the ode goes to zero as the size grows.

In opposition to Theorem 1, this theorem allows to restrict the approximation to a prefix of the trajectories, while beyond the exit time $\zeta(S)$ one can say nothing. This relaxed assumption allows to find piece-wise deterministic approximations [7] (called hybrid limits therein) also for PCTMC sequences that do not satisfy the assumptions of Theorem 1. However, Theorem 2 ensures a weaker form of convergence than Theorem 1, since almost sure convergence implies convergence in probability [5].

In both theorems nothing is said about asymptotic behaviour. This is a relevant topic, that allows to perform several studies such as steady state analysis of the population models as well as model checking [8]. In [3] the reader can find

a discussion on conditions under which one can draw conclusions also on the behaviour for T equal to ∞.

As a further remark we want to point out that Theorems 1 and 2 allow to establish that, in the limit, the error of deterministic approximation goes to zero. However, we are not able to *quantify* the error committed considering an intermediate system size. Details on worst-case bounds on this error can be found in [23]. A detailed proof of Theorem 2 can be found in [17,18].

3.3 Fast Simulation and Fluid Model Checking

An interesting consequence of the mean-field approximation theorem is the so-called *decoupling of joint probability* (for details, please refer to [3,36]). Let $\mathbf{S}^{(i)}(t)$ be the (parameterized) state of the system at time t, where $\mathbf{S}_k^{(i)}(t) \in \{1, \ldots, n\}$ is the state of the k-th object, and $\mathbf{S}_k(t)$ be the state of k in the limit model. Then, for any set of agents $1, \ldots, h$ and states $s_1, \ldots, s_h \in \{1, \ldots, n\}$, for large i:

$$\mathbb{P}\{\mathbf{S}_1^{(N)}(t) = s_1, \ldots, \mathbf{S}_h^{(N)}(t) = s_h\} \approx \mathbb{P}\{\mathbf{S}_1(t) = s_1\} \cdot \ldots \cdot \mathbb{P}\{\mathbf{S}_h(t) = s_h\}$$

That is, in the limit the joint probability distribution of the states becomes equal to the (product of the) independent probabilities of the states of the single agents. Therefore, we can approximate a single probability using the ODE solution as follows: $\mathbb{P}\{\mathbf{S}_1(t) = s_1\} = \overline{\mathbf{x}}_i(t)$. This holds because the limit is deterministic and the objects are abstracted w.r.t. their identities. However, since the mean-field approximation theorems hold for finite time horizon, we have no guarantee on the validity of decoupling also in the steady state, for $T = \infty$.

The decoupling of probabilities is a relevant property in many applications such as fast simulation [18,20] and fluid model checking [8]. The central idea of fast simulation is to abstract the system into its fluid approximation and to study the evolution of a single agent (or a fixed set of gents) as executed in parallel with the approximation. The advantage is that, rather than considering/simulating the entire system, it is sufficient to consider the abstract average behaviour of the system and observe a single agent interacting with it, by decoupling its evolution from the evolution of the remaining agents. This is a faithful approximation since, by Theorems 1 and 2 the dynamics of a single agent depend on the other agents only through the global system state.

This idea is further exploited in fluid model checking [8], where one studies *properties of a single agent* in time, within a large population. In particular, fluid model checking takes advantage of fluid approximation to obtain a more efficient stochastic model checking technique [35]. In [8] the authors develop novel CSL model checking algorithms for ICTMC models and show how to exploit fast simulation in this setting.

In this tutorial we illustrate an application of this technique in Section 6 by considering the system that we describe in the following Section 4.

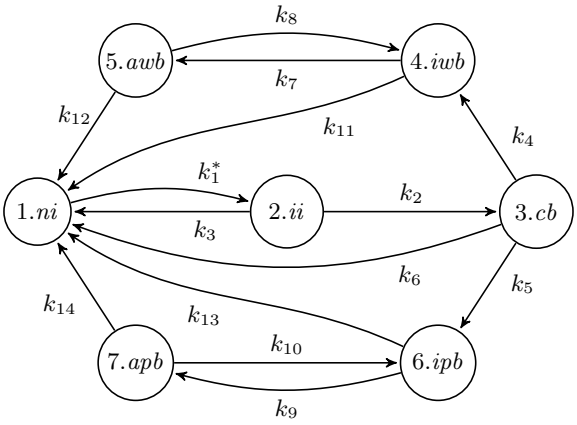

Fig. 2: Possible states of a computer in the network. The shorthand names are defined as follows: *ni=NotInfected*, *ii=InitialInfection*, *cb=ConnectedBot*, *iwb=InactiveWorkingBot*, *awb=ActiveWorkingBot*, *ipb=InactivePropagationBot*, and *apb=ActivePropagationBot*.

4 Mean-field Analysis of a Bot-net

In this section we discuss the applicability of the mean-field method to modeling peer-to-peer botnet, similarly to [31] . In Section 4.1 we discuss the characteristics of the botnet, which are important for modeling. Section 4.2 describes the mean-field model of the botnet spread. The performance evaluation results are presented in Section 4.3, together with an example of wider usability of the mean-field model.

4.1 Description of the system

Let us describe the steps each computer goes through during the botnet spread. The computer which is in *NotInfected* state (S_1) enters the *InitialInfection* (S_2) state with rate k_1^*. Then, it connects to the other bots in the botnet, going to *ConnectedBot* state (S_3), and it downloads the program containing the malware with rate k_2. If the computer, for some reason, is not able to download the malware, it returns to the state *NotInfected* with rate k_3.

After downloading the malware, the computer joins the botnet either as *InactiveWorkingBot* (S_4) or as *InactivePropagationBot* (S_6) with rates k_4 and k_5, respectively. If downloading the malware is not possible, for example, because the connection has failed, the computer moves back to the *NotInfected* state with rate k_6. Once the bot becomes either an *InactiveWorkingBot* or an *InactivePropagationBot* it never switches between the *Working-* or *Propagation-*classes. In order not to be detected, the bot is inactive most of the time and it only becomes active for a very short period of time. Transitions from *InactivePropagationBot* to *ActivePropagationBot* (S_7) and back occur with rates k_9 and

k_1	RateOfAttack·ProbInstallInitialInfection
k_1^*	Rate depends on k_1 and the environment
k_2	RateConnectBotToPeers·ProbConnectToPeers
k_3	RateConnectBotToPeers·(1-ProbConnectToPeers)
k_4	RateSecondaryInjection·ProbSecondaryInjectionSuccess·(1-ProbPropagationBot)
k_5	RateSecondaryInjection·ProbSecondaryInjectionSuccess·ProbPropagationBot
k_6	RateSecondaryInjection·(1-ProbSecondaryInjectionSuccess)
k_7	RateWorkingBotWakens
k_8	RateWorkingBotSleeps
k_9	RatePropagationBotWakens
k_{10}	RatePropagationBotSleeps
k_{11}	RateInactiveWorkingBotRemoved
k_{12}	RateActiveWorkingBotRemoved
k_{13}	RateInactivePropagationBotRemoved
k_{14}	RateActivePropagationBotRemoved

Table 1: Transition rates for a single computer.

k_{10}, respectively. The transition rates for moving from *InactiveWorkingBot* to *ActiveWorkingBot* (S_5) and back are denoted k_7 and k_8, respectively.

The computer can recover from its infection, e.g., if an anti-malware software discovers the virus, or if the computer is physically disconnected from the network. In these cases, it leaves the *InactivePropagationBot* or the *ActivePropagationBot* state and moves to the *NotInfected* state with rates k_{13}, k_{14}, respectively. The same holds for the working bots: the transition rates from *InactiveWorkingBot* and *ActiveWorkingBot* are k_{11}, k_{12}, respectively.

The model we construct considers several computers in a network, each of them being in one of the above mentioned states $S_1, .., S_7$, depicted also in Figure 2. The rates of transitions between states may depend on several factors, e.g., probability of a successful connection between initially infected computer and another infected computer, while moving from the state *InitialInfection* to the *ConnectedBot* state; or the probability of *ConnectedBot* to become *Porking* or *Propagation* bot, respectively. Table 1 provides the description of the transition rates for one computer model, while numerical values are given in Table 2. Rates $k_2 \dots k_{14}$ are constant for each computer, while rate k_1^* to move from the *NotInfected* state (S_1) to the *InitialInfection* state (S_2) is not constant. This rate depends on k_1 and on the number of computers in the *ActivePropagationBot* state, which are responsible of spreading the malware.

4.2 Mean-field Model

We study the spread of the botnet in a network of N computers by using the mean-field approximation method for finding the (average) deterministic dynamics of the system. The mean-field model captures the number of objects in a particular state, rather than considering the state of each single object. The mean-field state vector $\mathbf{X} = \langle X_1, X_2, \dots X_7 \rangle$ counts how many computers are in states $S_1, ..., S_7$. The occupancy measure is found by normalizing $\mathbf{X}$ into $\overline{\mathbf{X}}$.

We first construct the rate matrix, which collects the rates with which possible transitions take place. Transition rates may depend on time as well as on the state $\overline{\mathbf{X}}(t)$ of the system. The rate matrix $\mathbf{R}(\overline{\mathbf{X}}(t))$ of the model is given as:

$$\mathbf{R} = \begin{pmatrix} 0 & k_1^*(\overline{\mathbf{X}}(t)) & 0 & 0 & 0 & 0 & 0 \\ k_3 & 0 & k_2 & 0 & 0 & 0 & 0 \\ k_6 & 0 & 0 & k_4 & 0 & k_5 & 0 \\ k_{11} & 0 & 0 & 0 & k_7 & 0 & 0 \\ k_{12} & 0 & 0 & k_8 & 0 & 0 & 0 \\ k_{13} & 0 & 0 & 0 & 0 & 0 & k_9 \\ k_{14} & 0 & 0 & 0 & 0 & k_{10} & 0 \end{pmatrix} \tag{1}$$

The $|\mathbf{X}|\times|\mathbf{X}|$ infinitesimal generator matrix $\mathbf{Q}(\overline{\mathbf{X}}(t))$ is given as follows: $\mathbf{Q}(\mathbf{s}_1, \mathbf{s}_2)$ is equal to the transition rate $\mathbf{R}(\mathbf{s}_1, \mathbf{s}_2)$ to move from the state $\mathbf{s}_1$ to the state $\mathbf{s}_2$ and $\mathbf{Q}(\mathbf{s}, \mathbf{s})$ is equal to the reciprocal of the sum of all the rates in row $\mathbf{s}$. In a given example the only rate which depends on a state of the system is the infection rate $k_1^*(\overline{\mathbf{X}}(t))$, which depends on the number of computers (bots) actively spreading infection. The total rate of infections produced by all bots that are in the active propagation state is $k_1 \cdot \overline{X}_7(t)$. These infections are spread out randomly over all not-yet infected computers, whose number is denoted by $\overline{X}_1(t)$ [5]. Hence, the infection rate k_1^* perceived by each individual computer is given by the ratio:

$$k_1^*(\overline{\mathbf{X}}(t)) = \frac{k_1 \cdot \overline{X}_7(t)}{\overline{X}_1(t)}. \tag{2}$$

which entails that $\mathbf{Q}$ satisfies density dependence, as given in Definition 4.

One we have constructed the infinitesimal generator matrix $\mathbf{Q}$, we can use it to construct the set of Ordinary Differential Equations whose solution represents the average dynamics of the system. In particular, the drift matrix F is exactly the matrix $\mathbf{Q}$. The state vector on the continuous state space is $\mathbf{x} = \langle x_1, \ldots, x_7 \rangle$. Therefore, the initial value problem we study is defined as follows:

$$\frac{d\,\mathbf{x}(t)}{dt} = \mathbf{x}(t)\mathbf{Q}(\mathbf{x}(t)), \qquad \text{with initial condition } \mathbf{x}(0). \tag{3}$$

The system of equations we obtain is:

$$\begin{cases} \dot{x}_1(t) &= k_3 x_2(t) + k_6 x_3(t) + k_{11} x_4(t) \\ & \quad + k_{12} x_5(t) + k_{13} x_6(t) + (k_{14} - k_1) x_7(t) \\ \dot{x}_2(t) &= -(k_2 + k_3) x_2(t) + k_1 x_7(t) \\ \dot{x}_3(t) &= k_2 x_2(t) - (k_4 + k_5 + k_6) x_3(t) \\ \dot{x}_4(t) &= k_4 x_3(t) - (k_7 + k_{11}) x_4(t) + k_8 x_5(t) \\ \dot{x}_5(t) &= k_7 x_4(t) - (k_8 + k_{12}) x_5(t) \\ \dot{x}_6(t) &= k_5 x_3(t) - (k_9 + k_{13}) x_6(t) + k_{10} x_7(t) \\ \dot{x}_7(t) &= k_9 x_6(t) - (k_{10} + k_{14}) x_7(t) \end{cases} \tag{4}$$

[5] In the considered example the propagation bots are "smart" enough to spread infection via not infected computers only.

Parameter	Experiments		
	Baseline	Exper 1	Exper 2
ProbInstallInitialInfection	0.1	**0.06**	**0.04**
ProbConnectToPeers	1	1	1
ProbSecondaryInjectionSuccess	1	1	1
ProbPropagationBot	0.1	0.1	0.1
RateOfAttack	10.0	10.0	10.0
RateConnectBotToPeers	12.0	12.0	12.0
RateSecondaryInjection	14.0	14.0	14.0
RateWorkingBotWakens	0.001	0.001	0.001
RateWorkingBotSleeps	0.1	0.1	0.1
RatePropagationBotWakens	0.001	0.001	0.001
RatePropagationBotSleeps	0.1	0.1	0.1
RateInactiveWorkingBotRemoved	0.0001	0.0001	0.0001
RateActiveWorkingBotRemoved	0.01	0.01	0.01
RateInactivePropagationBotRemoved	0.0001	0.0001	0.0001
RateActivePropagationBotRemoved	0.01	0.01	0.01

Table 2: Setup for the three experiments. Bold indicates differences w.r.t. baseline.

The equations can be solved analytically, however the closed forms are impractically large. We used Wolfram Mathematica [45] to obtain the analytical solution.

4.3 Results

In this section we discuss the mean-field results in detail and compare them to the simulation results, the chosen parameters for all these experiments are given in Table 2. We essentially experimented considering different infection rates, denoting possible user behaviors, and their impact on the system behavior.

The simulation of the model was done using the Moebius tool [19] as in [44]. Each experiment covered one week of simulated time. Each experiment was replicated 1000 times; the mean values and 95% confidence intervals of the measures of interest are shown. The initial conditions for each experiment are as follows: 200 computers are located in the place *ActivePropagationBots*.

We use Mathematica [45] to obtain solutions for the set (4) of differential equations coupled with the transition rates from Table 2. Given an overall population of $N = 10^7$, the fraction of computers in the state *NotInfected* is initialized as $x_1(0) = (N - 200)/N$, the fraction of computers in the state *ActivePropagationBot* is initialized as $x_7(0) = 200/N$, and the fractions of computers in all other states are initialized as zero.

Figure 3 shows the number of the propagation bots along time. The number of propagation bots (both active and inactive) has been taken as measure of interest since they actively infect "healthy" computers. A logarithmic scale has been chosen for the number of propagation bots, in order to better visualize the exponential growth. The figure depicts the mean-field results of the Baseline ex-

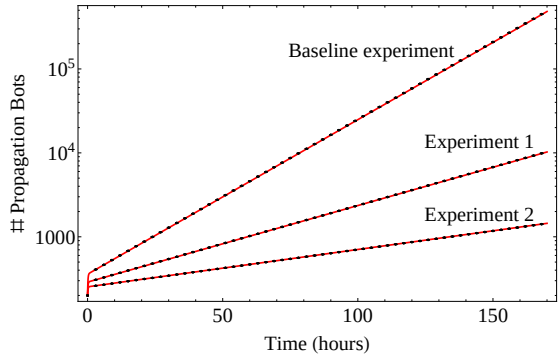

Fig. 3: Number of propagation bots over time in the Baseline experiment and experiments 1 ad 2 obtained from mean-field approximation together with the confidence intervals obtained from the simulation.

Experiment	Simulation	Mean-field
Baseline	5 d 3 h 25 min	1 sec
Exp. 1	9 h 51 min	1 sec
Exp. 2	5 h 37 min	1 sec

Table 3: Time spent on simulation and mean-field approximation.

periment together with the 95% confidence intervals of the Moebius simulation. As can be seen, the mean-field results are very accurate in this case, since they lie mostly within the confidence intervals, even though the confidence intervals are very narrow.

To investigate how a reduced infection spread would influence the growth of botnets, Experiments 1 and 2 were done in [44]. The "user factor" (*ProbInstalInfection*) is reduced to 60% and 40%, respectively, as compared to the Baseline experiment to represent a lower probability of, e.g., opening infected files. The results are, together with those from the Baseline experiment, presented in Figure 3. For both experiments, the results obtained with the mean-field model are very accurate and lie well within the confidence intervals most of the time.

One of the advantages of the mean-field method is that the time, needed for obtaining the means of the model is much smaller than the time, needed for the simulation (as shown in Table 3). The timings were obtained on a i7 processor with 3 GB RAM and 4 hyper-threading cores. The baseline experiment took 5 days 3 hours and 25 minutes, while the mean-field analysis was completed in one second. The difference between the simulation time for the different experiments is due to the dependency of the rates on a number of computers in *ActivePropagationBots* state. In the Baseline experiment the number of these computers is large, hence, the rate of infection becomes very large and more time is needed to simulate the resulting large number of events. The time spent on the simulation of the experiments with a lower number of computers involved is reasonably smaller; however the mean-field approximation is still much faster in all cases.

We do not provide all the experiments from [44] and [31] since they lie out of the scope of interest of this tutorial. Note, however, that the accuracy of the results and the speed of calculation hold for all the experiments, provided in the papers, mentioned above.

The speed of the mean-field results calculation allows us to use the mean field method to address problems which are not feasible using simulation: (i) we study the dependence of the botnet spread on two parameters, while the previous results are only functions of time for a given set of parameter values, (ii) and we study the behavior of the botnet in the presence of cost constraints. The purpose of the following is to show the difference between the simulation and the mean-field capabilities, and, at the same time, to show the advantages of the fast analysis.

We calculate the number of propagation bots as a function of k_{13} and k_{14} (see Figure 4). As one can see, there is no considerable difference in a relative increase of one or the other parameter. It is known that inactive computers are much harder to detect (increasing k_{13} is more difficult), therefore the above results might be helpful for the antivirus software developers to find the better strategy for botnet removal.

Next, we introduce a cost concept to analyze the economical side of an infection. Two types of costs are considered: (i) the cost of a computer being infected, for example due to the loss of information or productivity, and (ii) the cost of more frequent checking with antivirus software. On one hand the number of infected computers, and hence their cost grows if computers are not frequently checked. On the other hand, if computers are checked too often the botnet is not growing, but running the antivirus software becomes very expensive. We analyze this trade-off in more detail in the following. We calculate the cumulative cost between t_0 and t_1 as follows:

$$C(t_0, t_1, RR, D_1, D_2) \;=\; \int_{t_0}^{t_1} \left(D_1 \cdot \mathrm{IC}(t, RR) + D_2 \cdot RR \cdot \mathrm{AC} \right) dt \qquad (5)$$

where RR is the change in removal rates $k_{11}, ..., k_{14}$ with respect to the rates in the baseline experiment, i.e. $k_{11} = RR \cdot k_{11,baseline}$ (similarly for k_{12}, k_{13}, k_{14}); D_1 is the cost of infection; $\mathrm{IC}(t, RR)$ is the number of infected computers for a given RR, at time t, including active and inactive working and propagation bots; D_2 is the cost of one computer being checked, which probably is much lower than the cost of infection (D1); AC is the number of the computers in the network. We calculate the cumulative cost of the system performance for three days. For RR from the interval $[0.001; 5]$ we calculate the cost as a function of time for given D_1 and D_2. Results are depicted in Figure 5 and, one can see, that the cost grows exponentially with time and almost linearly with decreasing RR if the computers are not checked frequently (for the RR between 0 and 1). However, if antimalware software is used too often (RR above 2), the cost grows linearly with RR.

We see that the mean-field method can be easily used for finding the removal rates which minimize the cost at a given moment of time. It can help network managers with careful decision-making, based on the situation at hand. Even

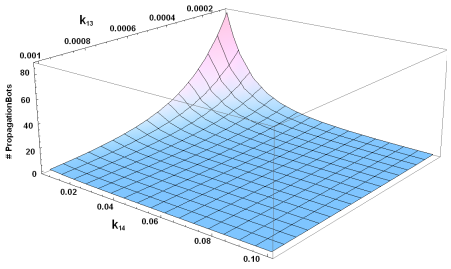

Fig. 4: Number of propagation bots for $(k_{13}, k_{14}) \in [8 \cdot 10^{-5}; 10^{-3}] \times [8 \cdot 10^{-3}; 10^{-1}]$ at time $T = 3days$, all other parameters are the same as for baseline experiment (see Table 2).

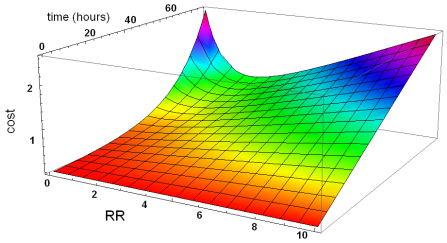

Fig. 5: Cost of the system performance for $D_1 = 0.01, D_2 = 4 \cdot 10^{-5}$.

though not all parameters might be known in reality, such analysis can help to obtain a better understanding of the characteristics of botnet spread.

In this section the basic mean-field example was described together with the possible extensive use of the mean-field model. An example of using mean-field approximation for more sophisticated systems is given in the next sections.

5 Spatial Mean Field Models

Early use of the mean-field analysis technique stems from the fields of physics (e.g. when studying gas dynamics) and systems biology (e.g. studying how concentrations of reactants behave in a solution). In those domains, the spatial distribution of particles/molecules across the system is not described in the model. Indeed, they assume that particles/molecules are uniformly spread across the space, thus ignoring the effect locations have on the overall dynamics. Systems where this assumption is realistic are often referred to as *homogeneous*, in physics, and *well-stirred*, in chemistry. In practice, this assumption implies that a single rate can be assigned for each type of particle-to-particle interaction, regardless of the spatial structure, and the interactions have the same probability to take place at any location.

In this section we focus on the appropriateness of this abstraction in the mean-field method, particularly in the context of modelling computer and communication networks. Depending on the nature of a given system, ignoring locations might be a suitable simplifying step. In our previous example, where we studied the spread of a virus in a network, the decision was made not to include the location of the computers. This led to a state vector ξ which only counted how many agents are in each of the states ni, ii, cb, iwb, awb, ipb and apb and the transition rate functions did not depend on distribution of computers across different geographical locations. Nevertheless, there exist systems whose dynamics and emergent behaviour are in fact, significantly dependent on locations. For such systems, if the model does not take into account such a spatial aspect,

the system behaviour may not be captured effectively. In such cases, the model should include an appropriate notion of agent location.

In this section, we consider an example of a large-scale peer-to-peer gossip network [14] where the emergent behaviour of the system significantly depends on locations. We describe, for this example, how the mean-field equations are constructed in a way that they also capture the effect that locations have on the system behaviour.

A second extension we present in this section concerns the application of the deterministic approximation theorem to *uncountable* domains. In Section 4 we illustrated an application of mean-field approximation to a finite-domain CTMCs. However, Kurts Theorem [33], as well as derived theorems (see Section 3), can be applied to Markov chains on countable domains [38]. The example considered in this section falls outside the scope of those results as it is applied on a Markov stochastic process on a *continuous* domain. Indeed, individuals (that is, taxis in this case) hold information concerning their location, that ranges on a finite set, and on the age of certain information they carry, that ranges on positive, real numbers. We will not address the technicalities related to this extension, but we point out this result, which in [14] is proved for the specific model considered and, in general, can not be obtained in a straightforward way. The uninterested reader can simply ignore this aspect and focus on the modelling of space.

5.1 The Age of Gossip

We consider the example from [14], which models a peer-to-peer communication network where two types of agent are present: some can move through different locations (mobile) and some others are stationary (base stations). The base stations transmit fresh updates on a piece of data through radio waves and these updates are received by the mobile agents. The data is time-stamped. The age of a piece of data on an agent is defined to be the time elapsed since last emission from a base station. The *age* of data received by an agent from a base station is *zero*. Agents are capable of radio communication between themselves. If two mobile agents get close enough, the agent who has the most recent version of data transmits the data to the other agent. The data exchange between two entities (a mobile agent receiving data from a base station or a mobile agent communicating with another mobile agent) takes place when the entities get close enough to establish a radio connection.

The system consists of a finite number of *locations* through which the agents can move. Each mobile agent can only be in one location at any time. The base stations in location c can establish radio communication only with agents who are in the same location. The data exchange between two mobile agents can take place either when the communicating agents both belong to the same location or when they are in two different locations. The latter type of communication captures, for example, the situation when two nodes from different locations are at the borders of adjacent locations and exchange data. We are interested into studying how, in each location, the age distribution of agents evolves over time.

A Formal Description. Let $L = \{1, 2, \dots C\}$ denote the set of locations and assume that there are N mobile agents who are moving across these locations. Let us define the variable X_i to represent the age of the i^{th} node and c_i to represent the location of node i. Hence, the state vector is $\boldsymbol{\xi} = \langle X_1, X_2, \dots X_N, c_1, c_2 \dots c_N \rangle$, $X_i \in \mathbb{R}_{\geq 0}$, and $c_i \in L$. We define the transitions and the rate function associated with each transition:

1. **Mobility.** A node can move from a location c to another location c' ($c, c' \in L$, $c \neq c'$) with rate $\rho_{c,c'}$. When there are N_c nodes in location c, the total rate at which nodes from location c move to location c' is $N_c \times \rho_{c,c'}$.
2. **Contact with base.** An agent i with age X_i in location $c \in L$ can communicate with a base station in location c and get fresh information. As the result of this data exchange, X_i becomes zero. For each location c a parameter μ_c describes the rate at which a node in location c can get information directly from base stations. If there is no base station in c, then $\mu_c = 0$.
3. **Opportunistic contact within location.** An agent i in location c communicates with another agent in the same location with rate $2\eta_c/(N-1)$. For each location c, there exists a parameter η_c, given by the modeller. This parameter is not dependent on the population of agents in that location. Even when two locations have the same population level, the rate at which the agents interact in those locations may not be the same. Indeed, the topological structure of c might encourage the agents to meet more frequently than c' and consequently, one will observe a higher interaction rate in c than in c'. The total interaction rate in location c is a function of both the population in that location (N_c) and η_c. Defining such a constant will particularly be useful when the modeller possesses real data about the execution of the system and wants to find parameters fitting the given data. If there are N_c nodes in location c, the total rate at which two nodes communicate is:

$$\binom{N_c}{2} \times \frac{2\eta_c}{(N-1)} = \frac{(N_c) \times (N_c - 1)}{N-1} \eta_c.$$

   This total rate includes the interaction of a node with nodes of any age.
4. **Opportunistic contact across locations.** A mobile agent in location c can communicate with a mobile agent from a neighbouring location c', ($c \neq c'$). This transition happens with rate $2\beta_{c,c'}/(N-1)$. For each c and c', ($c \neq c'$), $\beta_{c,c'}$ describes a constant which affects the rate at which the agents in c communicate with the agents in c'. The communication takes place only if there is at least one agent in c and one in c'.

State Space Representation - Choices. The location of each agent is one of its properties. For agent i, its location is in $L = \{1, 2, \dots C\}$. If we consider only this property of the agents, then the state vector would be $\boldsymbol{\xi}'(t) = \langle \xi'_1(t), \xi'_2(t), \dots \xi'_C(t) \rangle$ where for each location i, ξ'_i represents the population count at that location. Such population counts change over the course of time as the agents move between locations.

Let us assume that we use the state vector $\boldsymbol{\xi'}$ to model the peer-to-peer network and study how the system evolves. In the mean-field method, for each population count one differential equation is constructed. Therefore, given $\boldsymbol{\xi'}$, the system of differential equation will have C equations. The state space representation $\boldsymbol{\xi'}$ and the corresponding set of differential equations capture the evolution of agents only with respect to their locations. Using such a state representation, the other important property of the agents, i.e. their ages, is ignored.

Let us now consider how to model the other property of the agents, their age. The age of an agent can take values in $\mathbb{R}_{\geq 0}$. An agent has age zero if it has just had a communication with one of the base stations. The state of the system at time t can be characterized by a continuous distribution $\boldsymbol{\xi''}(z,t)$ with domain $\mathbb{R}_{\geq 0}$. $\xi''(j,t)$ captures how many agents have age (around) j at time t. Using the state representation $\boldsymbol{\xi''}(j,t)$, one can construct a set of *partial differential equations*, over the dimensions j and t, which captures how the agents evolve in terms of their age distribution as the time elapses. The shortcoming of this analysis is that the location of the agents, which has significant effect on how the age distribution evolves, is completely ignored.

In order to faithfully capture the dynamics of the considered system, a combination of both state representations $\boldsymbol{\xi'}$ and $\boldsymbol{\xi''}$ is needed, to consider both properties of the agents: their locations and their ages.

Mean Field State Space Representation. Consider a location c. For the i^{th} agent, who has age X_i, let us define the distribution δ_{X_i} which is a Dirac mass at X_i. At a time t, the age distribution of agents in location c across $\mathbb{R}_{\geq 0}$ is characterized by $M_c^N(t)$:

$$M_c^N(t) = \sum_{i=1}^{N} 1_{\{c_i=c\}} \delta_{X_i^N(t)}.$$

which is a continuous distribution denoting the number of agents who have age (around) z at location c and time t. The vector of continuous distributions

$$\boldsymbol{M}^N(t) = \langle\, M_1^N(\cdot,t), M_2^N(\cdot,t), \ldots, M_C^N(\cdot,t) \,\rangle$$

is defined in term d of the distributions $M_c^N(z,t)$, for each location $c \in L$, discussed above. This vector captures both location and age of an agent and is used, in the rest of this section, for mean-field analysis.

5.2 Mean-Field Limit Behaviour

In order to find the deterministic limit behaviour of the system, we first focus on the dynamics of the population moving across locations.

Mobility of the Agents. Let $\boldsymbol{U}(t) = \langle\, U_1(t), U_2(t), \ldots, U_C(t) \,\rangle$ be a vector such that $U_c(t)$ denotes the number of agents in location c at time t. The *location occupancy measure* is defined as:

$$\bar{\mathbf{U}}^N(t) = \frac{\boldsymbol{U}(t)}{N} = \langle \bar{\mathrm{U}}_0^N(t), \bar{\mathrm{U}}_1^N(t), \ldots, \bar{\mathrm{U}}_C^N(t) \rangle.$$

indicating the fraction of agents per location, at time t. Assume that, for $N \to \infty$, the sequence $\bar{\mathrm{U}}_c^N(0)$ converges to a unique limit (Definition 3):

$$\lim_{N\to\infty} \bar{\mathbf{U}}^N(0) = \lim_{N\to\infty} \frac{\boldsymbol{U}(0)}{N} = \left\langle \frac{U_1(0)}{N}, \frac{U_2(0)}{N}, \ldots, \frac{U_C(0)}{N} \right\rangle = \langle \bar{\mathrm{u}}_1^0, \bar{\mathrm{u}}_2^0, \ldots \bar{\mathrm{u}}_C^0 \rangle = \bar{\mathbf{u}}^0$$

Following [14], since convergence of initial occupancy measure holds and since constant mobility rates imply density dependence, we can apply Kurtz Theorem [34] (Theorem 1), and prove that *at any time* $t > 0$, for $N \to \infty$, the processe $\bar{\mathbf{U}}^N(t)$ converges to a deterministic process $\bar{\mathbf{u}}(t) = \langle \bar{\mathrm{u}}_1(t), \bar{\mathrm{u}}_2(t), \ldots \bar{\mathrm{u}}_C(t) \rangle$ where $\bar{\mathrm{u}}_c(t)_{c\in L}$ is the solution of the following initial value problem:

$$\begin{aligned} &\forall c \in L, \ \frac{\partial\ \bar{\mathrm{u}}_c(t)}{\partial t} = \left(\sum_{c' \neq c} \rho_{c',c} \bar{\mathrm{u}}_{c'} \right) - \left(\sum_{c' \neq c} \rho_{c,c'} \right) \bar{\mathrm{u}}_c \\ &\forall c \in L, \quad \bar{\mathrm{u}}_c(0) = \ \bar{\mathrm{u}}_c^0 \end{aligned} \tag{6}$$

The first term on the right hand side of Equation (6) indicates the increase of $\bar{\mathrm{u}}_c$ due to agents coming from adjacent locations. Similarly, the second term indicates the decrease of $\bar{\mathrm{u}}_c$ due to agents going towards adjacent locations.

According to [14], by the Cauchy-Lipschitz theorem, for any initial condition $\bar{\mathrm{u}}^0 = \langle \bar{\mathrm{u}}_c^0 \rangle_{c\in L}$, the above initial value problem admits a unique solution. $u_c(t \mid \bar{\mathrm{u}}^0)$ denotes the deterministic value of the location occupancy measure at time t given the initial condition $\bar{\mathrm{u}}^0$. In [14] the system behavior is studied at stationary mobility regime. For this purpose one can use the fixed point method:

$$\begin{aligned} &\forall c \in L, \frac{\partial\ \bar{\mathrm{u}}_c(t)}{\partial t} = 0 \quad \Rightarrow \\ &\forall c \in L, \tilde{u}_c \left(\sum_{c' \neq c} \rho_{c',c} u_{c'} \right) = \left(\sum_{c' \neq c} \rho_{c,c'} \right) \tilde{u}_c \ , \ \ \sum_{c \in C} \tilde{u}_c = 1 \end{aligned} \tag{7}$$

The solution of the above equation, $\tilde{u}$, shows how the agents are spread across the locations when the system reaches its equilibrium.

Propagation of Information - Age Distribution. Consider the state vector $\boldsymbol{M}$. For an agent population N and a time t, let us define the system's occupancy measure as a vector $\bar{\boldsymbol{M}}^N$ of continuous distributions:

$$\bar{\boldsymbol{M}}^N(t) = \frac{\boldsymbol{M}^N(t)}{N} = \langle\, \bar{M}_1(\cdot, t), \bar{M}_2(\cdot, t), \ldots, \bar{M}_C(\cdot, t)\, \rangle$$

For location c, $\bar{M}_c^N(z,t)$ denotes the *density* of agents in location c with age z at time t. For $\bar{M}_c^N(t)$, one can define its cumulative distribution function $F_c^N(z,t)$:

$$\forall c \in L, \ F_c^N(z,t) = M_c^N(t)[0:t] = \int_0^z \bar{\mathrm{M}}_c^N(s,t)\ ds$$

For location c, age z and time t, $F_c^N(z,t)$ tells us what proportion of the total population, at time t, is in class c with the age *less than or equal* to z.

We assume that, for $N \to \infty$, similarly to $\bar{\mathbf{U}}^N(0) \to \bar{\mathbf{u}}^0$, the vector of the occupancy measures $\bar{\mathbf{M}}^N(0)$ converges to a unique limit vector $\bar{\mathbf{m}}^0$:

$$\lim_{N\to\infty} \bar{\mathbf{M}}^N(0) = \bar{\mathbf{m}}^0$$

This means that for each location, the corresponding occupancy measure $\bar{\mathrm{M}}_c^N(0)$ converges to a unique limit distribution $\bar{\mathbf{m}}_c^0$ (Definition 3):

$$\forall c \in L \ , \ \lim_{N\to\infty} \bar{\mathrm{M}}_c^N(0) = \bar{\mathrm{m}}_c^0$$

As a consequence, at any given time $t > 0$ and for all $c \in L$, when N gets large, the density $\bar{\mathrm{M}}_c^N(t)$ converges to $\bar{\mathrm{m}}_c(t)$, where $\bar{\mathrm{m}}_c(t)$ is the solution of the following partial differential equation. In the following equation, $\bar{\mathrm{u}}_c(t)$ is the solution of Equation (6), the population of agents in location c at time t.

$$\bar{\mathrm{m}}_c(0,t) = \mu_c \times \bar{\mathrm{u}}_c(t) \tag{8}$$

$$\begin{aligned}
&\frac{\partial \bar{\mathrm{m}}_c(z,t)}{\partial t} = -\frac{\partial \bar{\mathrm{m}}_c(z,t)}{\partial z} - \mu_c \times \bar{\mathrm{m}}_c(z,t) \\
&+ \sum_{c'\neq c} \rho_{c',c} \bar{\mathrm{m}}_{c'}(z,t) - \left(\sum_{c'\neq c} \rho_{c,c'} \right) \bar{\mathrm{m}}_c(z,t) \\
&+2\eta_c \left[(+1) \times (u_c(t) - F_c(z,t)) \cdot \bar{\mathrm{m}}_c(z,t) + (-1) \times \bar{\mathrm{m}}_c(z,t) \cdot F_c(z,t) \right] \\
&+ \sum_{c'\neq c} 2\beta_{c,c'} \left[(+1) \times (u_c(t) - F_c(z,t)) \cdot \bar{\mathrm{m}}_{c'}(z,t) + (-1) \times \bar{\mathrm{m}}_c(z,t) \cdot F_{c'(z,t)} \right]
\end{aligned}$$

The formal proof of convergence is presented in [14]. However, here we use a more intuitive description, also presented in [14], to understand how the equations above are constructed.

Equation (8) can be formed by considering how much each $\bar{\mathrm{m}}_c(z,t)_{c\in C}$ changes in a small period of time ∂t (the left hand side). Consider location c. During ∂t, agents with age z, which have been accounted for by $\bar{\mathrm{m}}_c(z,t)$, will become older. Consequently, such agents need to be removed from $\bar{\mathrm{m}}_c(z,t)$. On the other hand, agents who currently have the age $z - \triangle z$ will become older and therefore, the density $\bar{\mathrm{m}}_c(z - \triangle z, t)$ will be added to $\bar{\mathrm{m}}_c(z,t)$. Hence, the rate of change of $m_c(z,t)$ caused only by *aging* is:

$$\lim_{\triangle z \to 0} \frac{\mid \bar{\mathrm{m}}_c(z - \triangle z, t) - \bar{\mathrm{m}}_c(z,t) \mid}{\triangle z} = \frac{\partial \bar{\mathrm{m}}_c(z,t)}{\partial z}$$

This is captured by the first term on the right hand side of Equation (8). The second term reflects the communication of agents, accounted by $\bar{\mathrm{m}}_c(z,t)$, with one of the base stations in their same location. Communicating with one of the base stations reduces the agent's age to zero and hence, such agents have to

be removed from $\bar{\mathrm{m}}_c(z,t)$. If there are $\bar{\mathrm{m}}_c(z,t)$ agents in location c, given that the rate of communication with a base station in c is μ_c, then, in a period of ∂t, $\mu_c \times \bar{\mathrm{m}}_c(z,t) \times \partial t$ of the agents will communicate with the base stations and hence, have to be removed from $\bar{\mathrm{m}}_c(z,t)$. The rate of the change is then calculated as $\mu_c \times \bar{\mathrm{m}}_c(z,t)$.

The third expression shows the flow of agents into $\bar{\mathrm{m}}_c(z,t)$ as a result of agents with age z moving from neighbouring locations c' into c, $(c \neq c')$. For a given c and c', such movement decreases $\bar{\mathrm{m}}_{c'}(z,t)$ and increases $\bar{\mathrm{m}}_c(z,t)$.The flow rate from $\bar{\mathrm{m}}_{c'}(z,t)_{c' \in L}$ into location $\bar{\mathrm{m}}_c(z,t)$ at time t is $\rho_{c,c'}\bar{\mathrm{m}}_{c'}(z,t)$. Similarly, the fourth term reflects the movement of some of the agents contained in $\bar{\mathrm{m}}_c(z,t)$ out of c into the adjacent locations. The flow rate is calculated similarly.

The fifth term has two parts. The first, $2\eta_c \times (u_c(t) - F_c(z,t)).\bar{\mathrm{m}}_c(z,t)$, shows the rate of flow into $\bar{\mathrm{m}}_c(z,t)$ because of agents with the age higher than z in c communicating with agents who have age z in the same location. When an agent of age higher than z communicates with an agent of age z, the age of the older one reduces to z. The total density of agents in location c at time t is $\bar{\mathrm{u}}_c(t)$ and the density of agents whose age is less than or equal to z is $F_c(z,t)$. Therefore, the density of agents with age higher than z in c is $(u_c(t) - F_c(z,t))$. The rate expression depends on the density of population of agents in c with age higher than z, the density of agents in c with the age z and additionally, on η_c.

The second part: $-2\eta_c \times (\bar{\mathrm{m}}_c(z,t)) \times F_c(z,t)$ shows the drift out of $\bar{\mathrm{m}}_c(z,t)$ as a result of agents with the age z in c communicating with the agents of lower age in the same location. The interpretation of the sixth term is similar, the difference being that it captures the communications which take place between the agents who belong to two different locations c and c' as opposed to a communication where two parties belong to the same location.

If we simplify the equation above and integrate over z, we obtain the following equation for $F_c(z,t)$:

$$
\begin{aligned}
&\forall c \in L : \frac{\partial\ F_c(z,t)}{\partial t} = \\
&-\frac{\partial\ F_c(z,t)}{\partial z} + \left(\sum_{c' \neq c} \rho_{c',c}\ F_{c'}(z,t)\right) - \left(\sum_{c' \neq c} \rho_{c,c'}\right) F_c(z,t) \qquad (9)\\
&+\ (u_c(t|d) - F_c(z,t))(2\eta_c F_c(z,t) + \mu_c) \\
&+\ (u_c(t|d) - F_c(z,t)) \sum_{c' \neq c} 2\,\beta_{c,c'} F_{c'}(z,t) \\
&\forall c \in L, \forall t \geq 0 : F_c(0,t) = 0 \\
&\forall c \in L, \forall z \geq 0 : F_c(z,0) = F_c(z)
\end{aligned}
$$

Note that this model relies on the assumption that the agents' movements do not depend on the information propagation scheme. Therefore, the set of ODEs (6) which capture the evolution of the location occupancy measure can be constructed and solved independently.

5.3 Solution of the Equations

Let us now describe how the solution of Equation (9) is obtained for the case where there is only *one location* in the system and assuming that when the system starts at $t = 0$, every agent has age zero.

The solution is found by introducing a change of variables. Let us define the space $\mathcal{A} = \{(x, y) \in \mathbb{R} \times \mathbb{R} \mid x \geq 0, x+y \geq 0\}$. The function $G(x, y) : A \to [0, 1]$ is defined as $G(x, y) = F(x, x+y)$. Therefore, in order to know $F(z, t)$ it is enough to calculate $G(z, t-z)$. For a function G defined as follows:

$$\frac{\partial G(x,y)}{\partial x} = \frac{\partial F(z,t)}{\partial z}\bigg|_{(x,x+y)} + \frac{\partial F(z,t)}{\partial t}\bigg|_{(x,x+y)}.$$

Rearranging the terms in Equation (9), we obtain:

$$\frac{\partial G(x,y)}{\partial x} = (1 - G(x,y))(2\eta\, G(x,y) + \mu) \qquad \text{for } \; G(0,y) = 0 \tag{10}$$

The assumption that at time $t = 0$, no gossip exists in the network leads to the conclusion that at any given time $z < t$ and $y = t - z > 0$. For an arbitrary value of $y \in \mathbb{R}^+$, let us define $g_y : x \mapsto G(x, y)$. Therefore:

$$\frac{\partial g_y(x)}{\partial x} = (1 - g_y(x))(2\eta g_y(x) + \mu) \qquad \text{for } \; g_y(0) = 0$$

By Cauchy-Lipschitz Theorem, this equation has a solution. Once the value of $g_y(x)$ for a given x is found, the corresponding $F(z, t)$ can be easily calculated.

Single Location - Analytical Solution. In this case, the ODE can analytically be solved and leads to the following solution:

$$F(z,t) = \begin{cases} 1 - \dfrac{2\eta + \mu}{2\eta + \mu e^{(\mu+2\eta)z}} & if \; z \leq t \\ 1 - \dfrac{2\eta + \mu}{2\eta + \frac{2\eta F(z-t,0)+\mu}{1-F(z-t,0)} e^{(\mu+2\eta)t}} & if \; z > t \end{cases} \tag{11}$$

Consider that above, we illustrated the reasoning behind the first case of the solution (when $z \leq t$). The second case ($z > t$), corresponds to the situation where in the initial configuration of the system some agents have age greater than zero. Therefore, at some time t, it is possible that some of the agents in the system have ages higher than t. The proportion of the agents who at time t have age $z > t$ depends on the proportion of the agents who had age at least $(z - t)$ in the system initial configuration.

Performance Evaluation of Peer-to-Peer Dynamics. In terms of performance, a well designed peer-to-peer opportunistic network should guarantee

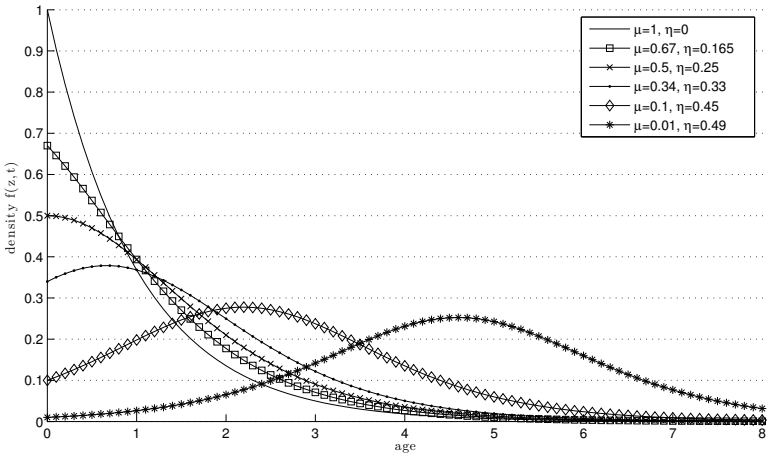

Fig. 6: the density at age z for different values of η and μ when $z \leq t$.

that with a high probability, the majority of agents remain within relatively low ranges of age. This performance requirement can be achieved adopting two different solutions: (1) increasing the frequency of contacts with base stations (which we identify as *infrastructure dominant* or (2) favoring interaction between mobile agents (which we identify as *opportunistic contact dominant*).

Figure 6 shows the results of the analysis of the model when the system consists of only one location. Different values for the parameters μ, η capture different degrees of dominance of the infrastructure or of the opportunistic contacts. We come to the following observations.

- **Infrastructure Dominant**. When $\mu \geq 2\eta$, the occupancy decreases as the age grows. The maximum density is at age $z = 0$ with $m(0, t) = \mu$. The rate at which opportunistic contacts take place is negligible with respect to the rate at which agents communicate with the base stations and hence, the latter type of communication determines the shape of the distribution. The extreme case, when $\eta = 0$, is the scenario where the opportunistic contact does not take place at all. In this case, improving the age distribution without changing the rate of the opportunistic contacts entails increasing the rate of communication with base stations.
- **Opportunistic Contact Dominant**. When $\mu < 2\eta$, the opportunistic contact rate becomes large enough to influence the age distribution. In such cases, there emerges a large mass around a *typical* age, which is maintained by the communication between the mobile agents. In the extreme case, μ is small and η is large. The mass around age $z = 0$ becomes negligible and depending on the frequency of the agent meetings, the dominant age is centered at some age $z > 0$. In order to improve the age distribution in such a network without changing μ, one needs to improve η which then leads to higher rates of agent-to-agent communication.

Multiple Locations. For the general case with multiple locations the solution method is more complicated [14]. Here we only explain the main ideas involved in the solution.

Let us assume that a sufficiently long time has elapsed since the initialization of the system and the distribution of agents across different locations has stabilized. That is, $\forall c \in L,\ \frac{\partial F_c(z,t)}{\partial t} = 0$ and $u_c(t)$ has converged to the equilibrium distribution $\tilde{u}_c$. Then, from Equation (9), we obtain:

$$\forall c \in L,\ \frac{d\ F_c(z)}{dz} = +\tilde{u}_c\mu_c + \left(\tilde{u}_c 2\eta_c - \mu_c - \sum_{c' \neq c} \rho_{c,c'}\right) F_c(z) \tag{12}$$
$$+ \sum_{c' \neq c} (\rho_{c',c} + \tilde{u}_c 2\beta_{c,c'}) F_{c'(z)} - \sum_{c' \neq c} 2\beta_{c,c'} F_c(z).F_{c'(z)} - 2\eta_c (F_c(z))^2$$
$$\forall c \in L,\ F_c(0) = 0$$

In contrast with the case of a single location, this ODE is multi-dimensional and has no simple analytical solution. However, we can distinguish the cases where the age is small and those where the age is large, thus finding a satisfactory linear approximation of (12). We now give more details about this approach.

For any location c, when $z \to 0$, $F_c(z)$ converges to zero. Hence, in Equation (12), the factors $F_c(z) \times F_{c'}(z)$ and $(F_c(z))^2$ become negligible compared to the rest of the expression and can be ignored. This approximation step will lead to the following system of equations which is shown in the matrix form:

$$F' = FA + B \tag{13}$$
$$A_{c,c} = \tilde{u}_c 2\eta_c - \mu_c - \sum_{c' \neq c} \rho_{c,c'}$$
$$A_{c,c'} = \rho_{c,c'} + \tilde{u}_{c'} 2\beta_{c,c'}$$
$$B = (\mu_0 \tilde{u}_0, \ldots, \mu_C \tilde{u}_C)$$

For location c and age z (z close to zero), the density of the nodes with that age is approximately $\mu_c \tilde{u}_c$. The derivative of the density function, $\frac{d\ \bar{\mathrm{m}}_c(z)}{dz}$ is:

$$\frac{d\ \bar{\mathrm{m}}_c(z)}{dz} = \mu_c \tilde{u}_c (\tilde{u}_c 2\eta_c - \mu_c - \sum_{c' \neq c} \rho_{c,c'}) + \sum_{c' \neq c} \mu_{c'} \tilde{u}_{c'} (\rho_{c',c} + \tilde{u}_c 2\beta_{c',c})$$

and if we assume $\forall\ c, c' \in L : \beta_{c,c'=0}$, then:

$$\frac{d\ \bar{\mathrm{m}}_c(z)}{dz} = \mu_c \tilde{u}_c (\tilde{u}_c 2\eta_c - \mu_c) + \sum_{c' \neq c} (\mu_{c'} - \mu_c) \tilde{u}_{c'} \rho_{c',c} \tag{14}$$

Equation (14) can be used to determine for a location c, whether c is a infrastructure dominant or opportunistic contact dominant. When for all locations c, $\mu_c = \mu$, i.e. when the base stations are distributed uniformly across different

locations, a location has a dominant infrastructure (respectively, dominant opportunistic contact) if $2\eta_c < \mu_c$ (respectively, $2\eta_c > \mu_c$). For the case when the base stations are installed in non-neighbouring locations, then a location with a base station has a dominant opportunistic contact if:

$$2\eta_c \tilde{u}_c > \mu_c + \sum_{c' \neq c} \rho_{c,c'}.$$

In every other location which does not have any base station, the age distribution will be dominated by the opportunistic contacts. The most general case happens when each location has its own specific μ_c and the base stations are distributed arbitrarily across the locations. In such a case, the nature of the location can be decided only after plugging the parameters into Equation (14) and observing the sign of the derivative at $z = 0$.

For the case when the modeller is interested in high values of age ($z \to \infty$), a similar technique can be used to simplified the equations [14].

5.4 Validation and Conclusions

The work in [14] proposes a stochastic model for the dissemination of timed stamped data in a spatial opportunistic peer-to-peer network. It illustrates how to model spatial aspects and how to adapt mean-field approximation in this context. Then, it considers real data and using classical stochastic simulations, it shows that the model is sound and sufficiently detailed. Finally, the authors illustrate how the mean-field approximation is accurate and much faster than simulation for this model. We now summarize the model validation steps and give hints on how realistic values for the parameters ρ, μ, β, and η are found.

CabSpotting [13] is a project of the company who runs the yellow taxi cabs in San Francisco Bay Area (SFBA). It consist in collecting in a database information about the location of each cab in the time period of one minute, recorded using GPS receivers.

The cabs in SFBA do not readily represent a peer-to-peer communication network. However, using the movement traces and considering realistic networking assumptions, one can construct a concrete spatial opportunistic peer-to-peer information dissemination network, similar to the model considered in Section 5.1. Using data from movement traces, one can extract the system parameters and feed them into the model. Such fully parametrized model can be analysed using a classical Monte Carlo analysis method, running a sufficiently large number of stochastic simulations. This allows to verify if the model faithfully captures the behaviour of the real system. These steps describe the approach taken in [14] for validation of the model. The outcome shows the model is sufficiently detailed to capture the real system dynamics. Then, the authors show that mean-field approximation are very accurate in describing how the age distributions in different locations evolve [14].

In the following two sections we review how the behaviour of a real opportunistic peer-to-peer network is constructed using the data in CabSpotting

database and how such a behaviour is used to extract the model's parameters: μ_c, η_c, $\rho_{c,c'}$ and $\beta_{c,c'}$.

Contact Traces. Assume the Bay Area is divided into 16 locations and some base stations are displaced through locations. Base stations transmit fresh data and have a specific transmission range. Each cab is equipped with a radio to communicate with base stations or other cabs, when sufficiently close. In [14], radio devices are assumed to have range of 200m. This is complies with standard radio technology used in vehicular networks. The taxi cabs scan their surrounding once per minute and upon detecting another entity (another cab or one of the base stations), they try to initiate a data exchange. A *meeting* or successful data exchange happens if the communicating agents remain in 200 meter proximity for at least 10 seconds (this duration guarantees data exchange). Under these specifications, one can generate *contact traces* which can be considered as the executions of a real spatial opportunistic peer-to-peer network and observe how the age distributions evolve in the real system. The cabs play the role of the mobile agents whose data is time stamped and the base stations are the sources for fresh information. In [14], contact traces were generated for dates between May, the 17th and June, the 15th, 2008 and for the time period between 8:00am till midnight, each day. Such traces were then used to calculate the age distributions at different time points and for different locations. The traces were also useful for finding parameters of the model and later for model validation. The validation process shows that for the locations which usually have reasonably large population of agents (having at least tens of taxi cabs), there exists a close correspondence between the age distributions obtained from the mean-field analysis of the model and the age distributions calculated by considering the contact traces. In the rest of this section, we look at the issue of how the model parameters are calculated based on the contact traces.

Extracting Model Parameters. Contact traces were used for calculating the following quantities:

1. $N(t)$: total number of cabs in time slot t (time unit = one minute).
2. $N_c(t)_{c\in\{1,2,3,\ldots 16\}}$: number of cabs in location c at time t.
3. $N_{c,ub}(t)$: number of contacts between a mobile node and a base station in location c at time t.
4. $N_{c,uu}(t)$: number of contacts between any two mobile nodes in location c at time t.
5. $N_{c,c',uu}(t)_{c\neq c'}$: number of contacts between an agent from c and another agent from c' at time unit t.

Given the contact traces, one can calculate $\mu_c(t) = \frac{N_{c,ub}(t)}{N_c(t)}$, which is the rate at which an agent in location c communicates with one of the base stations in that location. If at time t, there are $N_c(t)$ agents in location c, then on average,

one expects to observe $\mu_c(t) \times N_c(t)$ meetings in the following time unit. The average μ_c for an hour can be calculated by averaging $\mu_c(t)_{t\in[0:60]}$:

$$\mu_c = \frac{1}{60} \sum_{t=t_0}^{t_0+60} \mu_c(t).$$

Given the contact traces, for every location c, the parameter η_c is calculated by:

$$\eta_c(t) = \frac{N_{c,uu}(t)}{u_c(t) \times (N_c(t) - 1)}$$

The mean-field analysis assumed that in a location c, the rate at which an agents visits another agent in the same location is $\frac{2\times\eta_c}{N-1}$. Consequently, the rate at which one observe visits in location c is:

$$\binom{N_c}{2} \times \frac{2\eta_c}{(N-1)} = \frac{(N_c) \times (N_c - 1)}{N-1} \times \eta_c.$$

This means that on average, in one unit of time, we expect to observe $\frac{N_c\times(N_c-1)}{(N-1)}$ visits. On the other hand, the measurement from the simulations show that there have been observed $N_{c,uu}(t)$ visits in one time unit. Therefore:

$$N_{c,uu(t)} = \frac{(N_c) \times (N_c - 1)}{N-1} \times \eta_c \Rightarrow \mu_c = \frac{N_{c,uu}(t)}{\frac{N_c(t)}{N-1} \times (N_c(t) - 1)} \approx \frac{N_{c,uu}(t)}{u_c(t) \times (N_c(t) - 1)}.$$

The average η_c for one hour can be calculated by considering $\eta(t)$ for 60 minutes.

$$\eta_c = \frac{1}{60} \sum_{t=t_0}^{t_0+60} \eta_c(t)$$

Similarly, in the mean-field model, the rate at which an agent in location c visits an agent in location c' was assumed to be $\frac{2\times\beta_{c,c'}}{N-1}$. Therefore, in one time unit, on average $\frac{2\beta_{c,c'}}{N-1} \times N_c \times N'_c$ meetings occur between agents in location c and c'. The simulations show $N_{c,c',uu}(t)$ meetings having happened in time unit t. Therefore:

$$\frac{2\beta_{c,c'}}{N-1} \times N_c \times N_{c'} = N_{c,c',uu}(t) \Rightarrow \beta_{c,c'} = \frac{N_{c,c',uu}(t)}{2 \times N \times u_c \times N \times u_{c'} \times \frac{1}{N-1}} \Rightarrow$$

$$\beta_{c,c'} \approx \frac{N_{c,c',uu}(t)}{2 \times N(t) \times u_c(t) \times u_{c'}(t)}$$

hourly $\beta_{c,c'}$ can be calculated by averaging $\beta_{c,c'}(t)$ over an hour.

Finally, in the mean-field regime, the rate at which agents move from location c to c' is defined to be $\rho_{c,c'} \times N_c(t)$. In the simulations, one observes $N_{c,c',trans}(t)$ movements. Therefore:

$$\rho_{c,c'} \times N_c(t) = N_{c,c',trans}(t) \rightarrow \rho_{c,c'}(t) = \frac{N_{c,c',trans}(t)}{N_c(t)}.$$

The calculated parameters can then be used to built a fully parametrized model, which in turn can be used with different analysis method.

6 Fluid Model Checking

In Section 2 we discussed the relevance of the decoupling of probabilities for fast simulation [18,20] and fluid model checking [8]. Let us illustrate how this idea is used. Let $\mathbf{Z}_k^{(N)}(t) = \langle \mathbf{S}_1^{(N)}(t), \ldots, \mathbf{S}_k^{(N)}(t) \rangle$ be the state of k selected agents in the population, where k is fixed and independent of N. $\mathbf{Z}_k^{(N)}(t)$ is not an approximation, but it models exactly the dynamics of the agents, and it is not a CTMC, being the projection of the CTMC $\langle \mathbf{S}_1^{(N)}(t), \ldots, \mathbf{S}_N^{(N)}(t) \rangle$ on the first k coordinates, and may not be Markovian, in general. As a consequence, the limit of the model of a single agent has *rates depending on time*, i.e. it is a *time-inhomogeneous* CTMC (ICTMC). However, the entire process $\langle \mathbf{Z}_k^{(N)}(t), \overline{\mathbf{X}}^{(N)}(t) \rangle$ is Markovian. This kind of models allows to simulate the exact dynamics of a few agents in a large system very efficiently.

In [8], following the idea of fast simulation, the behaviour of the agent is singled-out as illustrated previously and their behavior is studied by considering their temporal properties. In particular, the evolution of $\mathbf{Z}_k^{(N)}(t)$ is model checked, while in parallel with $\overline{\mathbf{X}}^{(N)}(t)$. We know that $\mathbf{S} = \langle \mathbf{Z}_k^{(N)}(t), \overline{\mathbf{X}}^{(N)}(t) \rangle$ is Markovian while $\mathbf{Z}_k^{(N)}(t)$ is an ICTMC and we cannot reuse model checking algorithms for CTMCs. Therefore, in [8] the authors develop novel CSL model checking algorithms for ICTMC models and show how to exploit fast simulation in this setting. The overall system $\mathbf{S}$ satisfies Theorems 1 and 2 so the results of model checking are accurate for large populations.

In this section we discuss an application of the fluid model checking technique to population models. The kind of analysis we can perform through model checking is rather different from the performance studies we illustrated in Section 4. Indeed, we are able to formally prove temporal properties of the execution of these systems and have an estimate of the probability of their validity at a certain time point.

First, we illustrate a stochastic temporal logic (the bounded fragment of the CSL logic [2]) which we use to express those temporal properties. Then, we illustrate the algorithm to prove temporal properties of time-inhomogeneous CTMCs (ICTMC). The rates of the local ICTMC are approximated using fast simulation. Furthermore, we consider a simplified version of the example in Section 4 to illustrate the details of this technique and we prove some properties of interest.

6.1 Continuous Stochastic Logic

In the following, by $\mathcal{M}^l$ we indicate the model of $\mathbf{Z}_k^{(N)}(t)$. First, we recall the definition of *bounded* CSL [2]:

Definition 7. ***CSL Syntax.*** *Let $p \in [0,1]$ be a real number, $\bowtie \in \{\leq, <, >, \geq\}$ a comparison operator, $I \subseteq \mathbb{R}_{\geq 0}$ a non-empty bounded time interval and AP a set of atomic propositions with $a \in AP$. CSL* ***state formulas*** *Φ are defined by:*

$$\Phi ::= tt \mid a \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \mathcal{P}_{\bowtie p}(\phi),$$

where ϕ is a ***path formula*** *defined as:*

$$\phi ::= \mathcal{X}^I \Phi \mid \Phi_1 \; U^I \; \Phi_2.$$

To define the semantics of path formulas we first recall the notion of a path as in [2]. An *infinite path* σ is a sequence $s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} ...$ with, for $i \in \mathbb{N}$; $s_i \in S^l$ and $t_i \in \mathbb{R}_{>0}$ such that the probability that starting in state s_i we reach state s_{i+1} at time $t_\sigma[i] = \sum_{j=0}^{i} t_j$ is greater than zero. A finite path σ is a sequence $s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} ...s_{l-1} \xrightarrow{t_{l-1}} s_l$ such that s_l is absorbing, and, similarly, a probability of going from s_i to s_{i+1} is greater than zero for all $i < l$.

For a path σ, $\sigma[i] = s_i$ denotes for $i \in \mathbb{N}$ the $(i+1)st$ state of path σ. The time spent in state s_i is denoted by $\delta(\sigma; i) = t_i$. Moreover, with i the smallest index with $t \leq \sum_{j=0}^{i} t_j$, let $\sigma@t = \sigma[i]$ be the state occupied at time t. For finite paths σ with length $l+1$, $\sigma[i]$ and $\delta(\sigma; i)$ are defined in the way described above for $i < l$ only and $\delta(\sigma; l) = \infty$ and $\delta@t = s_l$ for $t > \sum_{j=0}^{l-1} t_j$. $Path^{\mathcal{M}^l}(s_i, t_0)$ is the set of all finite and infinite paths of the CTMC that start in state s_i given the state $\overline{x}$ at a certain time of the overall model $\mathcal{M}^l$ and $Path^{\mathcal{M}^l}(t_0)$ includes all (finite and infinite) paths of the CTMC, which depends on the overall system state (global time) if the CTMC is time-inhomogeneous. A probability measure[6] $Pr(t_0)$ on paths can be defined as in [2].

Since the local model changes with time, the satisfaction relation for a local state or path depends on time as well, and it is defined as follows:

Definition 8. ***Semantics of CSL.*** *Satisfaction of state and path CSL formulas for ICTMCs is given as follows:*

$$\begin{array}{ll}
s, t_0 \models tt & \forall s \in S^l, \\
s, t_0 \models a & \text{iff } a \in L(s), \\
s, t_0 \models \neg\Phi & \text{iff } s, t_0 \not\models \Phi, \\
s, t_0 \models \Phi_1 \wedge \Phi_2 & \text{iff } s, t_0 \models \Phi_1 \text{ and } s, t_0 \models \Phi_2, \\
s, t_0 \models \mathcal{P}_{\bowtie p}(\phi) & \text{iff } Prob^{\mathcal{M}^l}(s, t_0, \phi) \bowtie p, \\
\sigma, t_0 \models \mathcal{X}^I \Phi & \text{iff } \sigma[1] \in I, \text{ and} \\
 & \sigma[1], t_0 + t_{\sigma[1]} \models (\delta(\sigma, 0))\Phi \wedge \delta(\sigma, 0) \in I, \\
\sigma, t_0 \models \Phi_1 \; U^I \; \Phi_2 & \text{iff } \exists t' \in I : (\sigma@t' \models \Phi_2) \\
 & \wedge (\forall t'' \in [0, t')(\sigma@t'' \models \Phi_1)),
\end{array}$$

$I \subseteq \mathbb{R}_{\geq 0}$ is a non-empty time interval and $Prob^{\mathcal{M}^l}(s, t_0, \phi)$ is the probability measure of all paths $\sigma \in Path^{\mathcal{M}^l}(s, t_0)$ that satisfy ϕ and starting in state s, that is, $Prob^{\mathcal{M}^l}(s, t_0, \phi) = Pr\{\sigma \in Path^{\mathcal{M}^l}(s, t_0) \mid \sigma, t_0 \models \phi\}$.

Note that only *bounded* time intervals are used in path formulas. This is motivated by the nature of results ensured by the approximation Theorems 1 and 2, which are valid only for finite-time horizons. The relaxation of this restriction is possible, but we will not discuss it this tutorial, see [9], and [32] for details.

[6] Note that probability measure was denoted, in the preliminaries, by $\mathbb{P}$.

The CSL operators can be nested according to Definition 7. Model-checking of the CSL formula is done by building the *parse tree* and computing the satisfaction set of the individual operators recursively (in a bottom-up fashion), as described in [2]. Note that satisfaction set of the CSL formula is defined as follows: Model-checking CSL formulas for ICTMCs is similar to model-checking these formulas for CTMCs. All time-independent CSL operators can be checked using standard methods (see [2]) due to the independence of the results on time. Therefore, model-checking these operators is not included in the following discussion.

The main challenge is in model-checking *time-dependent* operators: let us first recall how these formulas are checked for time-homogeneous models. Given an arbitrary time-homogeneous CTMC $\mathcal{A}$, the probability formula containing the interval next operator $\mathcal{P}_{\bowtie p}\mathcal{X}^{[t_1,t_2]}\Phi$ is usually checked by computing the next-state probability and by comparing it with the threshold p [2]. This is calculated as the probability that the next jump starts within the time interval $[t_1; t_2]$ and ends in a state that satisfies Φ.

The probability formula including interval until formula $\mathcal{P}_{\bowtie p}\Phi_1 U^{[t_1,t_2]}\Phi_2$ for an arbitrary time-homogeneous CTMC $\mathcal{A}$ is checked by computing the probability of taking a path satisfying the until formula and by comparing it to the threshold p [2]. Let us denote the states satisfying Φ_2 as goal states, and the set of such a states as $\mathbb{G} = [\![\Phi_2]\!]$, a set of states satisfying Φ_1 as safe states $\mathbb{S} = [\![\Phi_1]\!]$, and, similarly, a set of the unsafe states $\mathbb{U} = [\![\neg\Phi_1]\!]$ for the ease of notation. For model-checking CSL until formula, we need to consider all possible paths, starting in a safe state $s_1 \in S$ at the current time and reaching a goal state $s_2 \in G$ during the time interval $[t_1, t_2]$ by only visiting safe states on the way. We can split such paths in two parts: the first part models the path from the starting state s to a state $s_1 \in \mathbb{S}$ and the second part models the path from s_1 to a state $s_2 \in \mathbb{G}$ only via safe states. We therefore need two transformed CTMCs: $\mathcal{A}[\mathbb{U}]$ and $\mathcal{A}[\mathbb{U} \cup \mathbb{G}]$, where $\mathcal{A}[\mathbb{U}]$ is used in the first part of the path and $\mathcal{A}[\mathbb{U} \cup \mathbb{G}]$ is used in the second. In the first part of the path, we only proceed along safe states thus all unsafe states $s \in \mathbb{U}$ do not need to be considered and can be made absorbing. As we want to reach a $\mathbb{G}$ state via $\mathbb{S}$ states in the second part, we can make all unsafe and goal states absorbing, because we are done as soon as we reach such a state.

In order to calculate the probability for such a path, we accumulate the multiplied transition probabilities for all triples (s, s_1, s_2), where $s_1 \in \mathbb{S}$ and is reached before time t_1 and $s_2 \in \mathbb{G}$ and is reached within time $t_2 - t_1$. Note that this formula is valid only for time-inhomogeneous CTMC, where the time when system is observed does not matter.

$$\mathrm{Prob}^{\mathcal{A}}(s, \Phi_1 U^{[t_1,t_2]}\Phi_2) = \sum_{s_1 \models \Phi_1} \sum_{s_2 \models \Phi_2} \pi^{\mathcal{A}[\mathbb{U}]}_{s,s_1}(t_1) \cdot \pi^{\mathcal{A}[\mathbb{U}\cup\mathbb{G}]}_{s_1,s_2}(t_2 - t_1). \tag{15}$$

Hence, CSL until formulas can be solved as a combination of two reachability problems, as shown in Equation (15), namely $\pi^{\mathcal{A}[\mathbb{U}]}_{s,s_1}(t_1)$ and $\pi^{\mathcal{A}[\mathbb{U}\cup\mathbb{G}]}_{s_1,s_2}(t_2 - t_1)$ that can be computed by performing transient analysis on the transformed CTMCs.

In the following we discuss the model-checking procedures that allow us to solve the interval path formulas (until and next) for the random local object, i.e. ICTMC. The procedure for checking these operators for ICTMCs is similar to that for CTMCs discussed above. However, the probabilities to take a certain path have to be calculated differently, because the Markov chain is time-inhomogeneous.

6.2 Next state probability

Since the local mean-field model is a ICTMC the standard model-checking procedure is not applicable, therefore in the following we explain how to calculate the next state probability of the local model. Note that this probability is also changing with time, therefore not only the next state probability at a given time t_0 is of interest, but also the dependency of such probability measure on time the formula is checked. Another important difference bethween checking CSL formulas for CTMC and ICTMC is in the fact that the set of goal states can change with time. The later is mostly useful for checking nested formulas, where the timed behavior of the sub-formulas leads to changes in the satisfaction relation. In the following we address these differences and explain how a bounded CSL Next fomula can be checked for the local mean-field model.

We first describe how to calculate the next state probability for a given time t_0 $\mathrm{Prob}^{\mathcal{M}^l}(s, \mathcal{X}[t_1, t_2]\Phi, t_0)$, i.e., the probability to jump from the state s to the state, satisfying Φ, or goal state, withing time interval $[t_1, t_2]$. This probability can be find as follows:

$$\mathrm{Prob}^{\mathcal{M}^l}(s, \mathcal{X}^{[t_1,t_2]}\Phi_2, t_0) = \int_{t_0+t_1}^{t_0+t_2} q_{s,\mathbb{G}}(t) \cdot e^{-\Lambda(s,t_0,t)} dt, \tag{16}$$

where $q_{s,\mathbb{G}}(t) = \sum_{s' \in \mathbb{G}} Q_{s,s'}(t)$ is the rate of jumping from the current state s to the goal state s' at time t; and $\Lambda(s, t_0, t) = \int_{t_0}^{t} -Q_{s,s}(\tau)d\tau$ is the cumulative exit rate of state s between t_0 and t. The proof is straight forward and can be found in [12].

The next state probability can now be computed numerically in two ways: using Equation (16) or by transformation the above formula to the differential equation and solving this equation. The differential equations, which are more convenient and simplify the calculations, can be obtained as in [9]:

$$\begin{cases} \dot{P}(t) = q_{s,\mathbb{G}}(t) \cdot e - L(t), \\ \dot{L}(t) = -q_{s,s}(t), \end{cases} \tag{17}$$

where $P(t_0 + t_1) = 0$ and $L(t_0 + t_1) = \Lambda(t_0, t_0 + t_1)$. The above ODEs have to be integrated from time $t_0 + t_1$ to time $t_0 + t_2$.

As we discussed above, for checking CSL formulas the dependency of the next state probability on time $\overline{P}_s(t) = \mathrm{Prob}^{\mathcal{M}^l}(s, \mathcal{X}^{[t_1,t_2]}\Phi_2, t_0, t)$ is needed to be accessed. To find this dependency one has to either calculate integral (16) for all possible t_0, or use the differential equations (17) to define another system of the differential equations with t_0 as a independent variable:

$$\begin{cases} \dot{\overline{P}}_s(t) = q_{s,\mathbb{G}}(t+t_2)\cdot e{-}L_2(t) - q_{s,\mathbb{G}}(t+t_1)\cdot e{-}L_1(t) - q_{s,s}(t)\overline{P}_s(t), \\ \dot{L}_1(t) = -q_{s,s}(t) + q_{s,s}(t+t_1), \\ \dot{L}_2(t) = -q_{s,s}(t) + q_{s,s}(t+t_2), \end{cases} \tag{18}$$

where $L_1(t) = \Lambda(t, t{+}t_1)$ and $L_2(t) = \Lambda(t, t{+}t_2)$. Initial conditions are computed by solving Equation (17).

The set of goal states can be time-dependent $\mathbb{G}(t)$, which has to be taken into account while calculating the next state probability. It is done by solving the above equation piecewise. All the time points $T_1, T_2, ...T_k$ when the goal set is changing are found first, where $T_0 = t_0 + t_1$ and $T_{k+1} = t_0 + t_2$. Equation (18) is solved for each time interval $[T_i; T_{i+1}]$.

Note that for checking next formula one has to compare next state probability with the given threshold $p \in [0, 1]$, hence, equation $\overline{P}_s(t) = p$ has to have a finite number of solutions. In general, this doesn't always hold, therefore, the restrictions on the rate functions of the mean-field model have to be introduced in order to insure the finite number of such solutions. In particular, the rate functions must be a *piecewise real analytical functions*, as described and proved in [12].

6.3 Until formulas. Reachability probability

The core idea of CSL model-checking of until formulas as explained in the previous section remains unchanged for time-inhomogeneous CTMCs. However, due to time-inhomogeneity it is not enough to only consider the time duration, but the exact time at which the system is observed must be taken into account. Hence, we add time t' to the notation of a time-inhomogeneous reachability problem $\pi^{\mathcal{M}^l}_{s,s_1}(t', T)$ to denote that we start in state s at time t'.

A probability for an arbitrary until formula $\Phi_1 U^{[t_1,t_2]} \Phi_2$ to hold is then again calculated by computing two reachability problems on the transformed local models $\mathcal{M}^l[\mathbb{U}]$ and $\mathcal{M}^l[\mathbb{U} \wedge \mathbb{G}]$, respectively:

$$\text{Prob}^{\mathcal{M}^l}(s, \Phi_1 U^{[t_1,t_2]} \Phi_2, t') = \sum_{s_1, t' \models \Phi_1} \sum_{s_2, t_1 \models \Phi_2} \pi^{\mathcal{M}^l[\mathbb{U}]}_{s,s_1}(t', t_1{-}t') \cdot \pi^{\mathcal{M}^l[\mathbb{U}\wedge\mathbb{G}]}_{s1,s2}(t_1, t_2{-}t_1). \tag{19}$$

Note that Equation (19) is valid for $t_1 > t', t_2 > t'$. If $t_1 = t'$ the first reachability problem can be omitted.

The standard transient analysis on the modified ICTMS is used in order to calculate the reachability probability $\Pi'(t', t' + T)$. In order to find the transient probability the forward Kolmogorov equation is solved with an identical matrix as initial condition:

$$\frac{d\Pi'(t', t'+T)}{d(T)} = \Pi'(t', t'+T) \cdot Q'(t'+T), \tag{20}$$

where $Q'(t'+T)$ is the rate matrix of the modified ICTMC.

In order to check CSL formula for ICTMC the dependency of transient probability on the starting time has to be found. The later is done by combining the forward and backward Kolmogorov equations:

$$\frac{d\Pi'(t,t+T)}{dt} = -Q'(t)\Pi'(t,t+T) + \Pi'(t,t+T)Q'(t+T). \tag{21}$$

Finally, the time-dependent probability matrix $\Pi'(t,t+T)$ can be obtained by solving Equation (21) with initial condition $\Pi'(t',t'+T)$. This can be done either analytically or numerically, e.g. with the tool Wolfram Mathematica [45] as used in the current paper. Note that using Kolmogorov equations for solving reachability problems on the local models $\mathcal{M}^l$ is efficient due to the fact that the state space is usually quite small (see [9]).

The goal and unsafe sets in ICTMC can vary with time, which has to be taken into account while calculating reachability probability. This is done by solving Equation (21) piecewise, i.e., for each time interval, where the above mentioned sets remain unchanged. At first we find the so-called discontinuity points, i.e., the time points $T_0 = t' \leq T_1 \leq T_2 \leq \cdots \leq T_k \leq T_{k+1} = T + t'$, where at least one of the sets changes. Then we do the integration separately on each time interval $[T_i, T_{i+1}]$ for $i = 0, ..., k$.

To ensure that only safe states are visited before a goal state is reached, we need to modify the CTMC $\mathcal{M}^l$ for each time interval as follows. First we introduce a new goal state s^*, which remains the same for all time intervals. Then, all unsafe and goal states are made absorbing and all transitions leading to goal states are readdressed to the new state s^*. Given this modified CTMC $\overline{\mathcal{M}}^l$, the transient probability matrix $\overline{\Pi'}(T_i, T_{i+1})$ is found for each time interval using the forward Kolmogorov equation, according to Equation (20).

Upon "jumps" between time intervals $[T_{i-1}, T_i]$ and $[T_i, T_{i+1}]$ it is possible that a state that was safe in the previous time interval becomes unsafe in the next. In this case the probability mass in this state is lost, since this path does not satisfy the reachability problem anymore. In the case that a state remains safe or a safe state is turned into a goal state the probability mass has to be carried over to the next time interval. This is described by the matrix $\zeta(T_i)$ of size $(|S^l|+1) \times (|S^l|+1)$ constructed in the following way: for each state $s \in S^l$ which are safe before and after T_i it follows $\zeta(T_i)_{s,s} = 1$. For each state $s \in S^l$ which was safe before T_i and become goal after T_i we have $\zeta(T_i)_{s,s*} = 1$. For the new goal state s^* the entry always equals one ($\zeta(T_i)_{s*,s*} = 1$), and all other elements of $\zeta(T_i)$ are 0.

The probability to reach a goal state before time T has passed when starting in a safe state at time t' is given then by the matrix $\Upsilon(t', t'+T)$:

$$\begin{aligned}\Upsilon(t', t'+T) =& \overline{\Pi'}(t', T_1) \cdot \zeta(T_1) \cdot \overline{\Pi'}(T_1, T_2) \cdot \\ & \zeta(T_1) \ldots \zeta(T_k) \cdot \overline{\Pi'}(T_k, t'+T).\end{aligned} \tag{22}$$

The probability to reach the goal state s^* is unconditioned on the starting state by adding 1 for all goal states:

$$\pi_{s,s^*}^{[\mathbb{U}\vee,\mathbb{G}]}(t',t'+T) = \Upsilon_{s,s^*}(t',t'+T) + \mathbf{1}\{s \in Sat(\mathbb{G},t')\}. \tag{23}$$

Similarly to the dependency on time of the reachability probability while the goal and unsafe sets are fixed (see Equation (21)), the time-dependent reachability probability for varying goal and unsafe sets can be found by again combining forward and backward Kolmogorov equations using chain rule.

The method for checking state and path CSL formulas was presented above in this section. As a next step we provide the example, where these methods are applied.

6.4 Examples

In this section some examples of checking CSL formulas are described. We use the model, similar to the botnet model, described in Section 4. In this model the number of possible states one computer goes through is reduced in order to simplify the reasoning and make the example more understandable.

The computer virus model, which is used as a running example in this section includes three possible modes of an individual computer, which can be *not-infected*, *infected and active* or *infected and inactive*. An infected computer is *active* when it is spreading the virus and *inactive* when it is not. This results in the finite local state space $S^l = \{s_1, s_2, s_3\}$ with $|S^l| = K = 3$ states. They are labelled as *infected*, *not infected*, *active* and *inactive*, as indicated in Figure 7. Transitions are similar to the botnet example, esplained in Section 4.

The system of ODEs (3), that describes the mean-field model of the computer virus is as follows:

$$\begin{cases} \dot{x}_1(t) = -k_1 x_3(t) + k_2 x_2(t) + k_5 x_3(t), \\ \dot{x}_2(t) = (k_1 + k_4) x_3(t) - (k_2 + k_3) x_2(t), \\ \dot{x}_3(t) = k_3 x_2(t) - (k_4 + k_5) x_3(t). \end{cases} \tag{24}$$

The coefficients that are used in the following example are given in Setting 1 in Table 4.

Let us consider the following formula

$$\Phi = \mathcal{P}_{<0.3}(\text{not infected } U^{[0,1]} \text{ infected})$$

and a predefined initial occupancy vector $\overline{x} = (0.8, 0.15, 0.05)$ at time $t' = 0$.

The only time-dependent rate of the local model is $k_1^*(t) = k_1 \cdot \frac{x_3(t)}{x_1(t)}$, where $m_1(t)$ and $x_3(t)$ are the solution of the ODEs (24) with $\overline{x}$ as initial condition. Therefore the transition rate matrix $\mathbf{Q}(t)$ equals

$$\mathbf{Q}(t) = \begin{pmatrix} -k_1 \cdot \frac{x_3(t)}{x_1(t)} & k_1 \cdot \frac{x_3(t)}{x_1(t)} & 0 \\ k_2 & -k_2 - k_3 & k_3 \\ k_5 & k_4 & -k_5 - k_4 \end{pmatrix}.$$

To find $Prob^{\mathcal{M}^l}(s, \text{not infected } U^{[0,1]} \text{ infected}, t')$ the reachability problem $\pi^{\mathcal{M}^l[\neg \text{not infected} \vee \text{infected}]}_{s,s_1}(0,1) = \pi^{\mathcal{M}^l[\text{infected}]}_{s,s_1}(0,1)$ has to be solved according to the algorithm described earlier in this section. The local model $\mathcal{M}^l$ is modified and all *infected* states are made absorbing. The Kolmogorov equation is used to calculate the transient probability matrix of the modified model, which consists of the reachability probabilities:

$$\Pi'(0,1) = \begin{pmatrix} 0.91 & 0.09 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The probability of the until formula

$$\phi = \text{not infected } U^{[0,1]} \text{ infected}$$

to hold for each starting state is as follows:
$Prob^{\mathcal{M}^l}(s_1, \phi, t') = \pi^{\mathcal{M}^l[\text{infected}]}_{s_1,s_2}(0,1) + \pi^{\mathcal{M}^l[\text{infected}]}_{s_1,s_3}(0,1) = 0.09$; $Prob^{\mathcal{M}^l}(s_2, \phi, t')) = 0$; $Prob^{\mathcal{M}^l}(s_3, \phi, t')) = 0$. As one can see the formula $\mathcal{P}_{<0.3}(\text{not infected } U^{[0,1]} \text{ infected})$ holds for all states s_1, s_2, and s_3.

As was discussed earlier, the satisfaction on the CSL formula may change with time. Let us consider the same formula $\mathcal{P}_{<0.3}(\text{not infected } U^{[0,1]} \text{ infected})$ and initial occupancy vector $\overline{x} = (0.8, 0.15, 0.05)$. In the following we calculate the time-dependent probability on the predefined time interval $[0, 20]$. The calculation of the time-dependent probabilities $Prob^{\mathcal{M}^l}(s, \text{not infected } U^{[0,1]} \text{ infected}, t', t)$ is done as described earlier in this section. The model $\mathcal{M}^l$ is modified so the infected states are made absorbing. The transient probability $\Pi(0,1)$ is calculated as described above. Forward and backward Kolmogorov equations are used in order to construct the ODEs, describing the time-dependent transient probability of the modified model (see Equation (21)). These ODEs are solved using $\Pi(0,1)$ as initial condition. The solution of the ODEs defines the required reachability probabilities. The probabilities $Prob^{\mathcal{M}^l}(s, \text{not infected } U^{[0,1]} \text{ infected}, t', t)$ are calculated by combining reachability probabilities (in this case equals to the reachability probabilities, which were calculated above). The time-dependent probability $Prob^{\mathcal{M}^l}(s_1, \text{not infected } U^{[0,1]} \text{ infected}, t', t)$ is depicted in Figure 8. Starting at states s_2 and s_3 this probability equals zero at all times, since these states do not satisfy *not infected*. In order to find the satisfaction set of this formula the following equation $Prob^{\mathcal{M}^l}(s_1, \text{not infected } U^{[0,1]} \text{ infected}, t', t) = 0.3$ is solved and $t = 13.42$ is found. The satisfaction set depends on time and includes all three states s_1, s_2, and s_3 for $t \in [0, 13.42)$; and only two states s_2 and s_3 for $t \in [13.42; 20]$.

In the following we discuss a more involved example, which includes nested until formula. The parameters of the model used in this example are given in the column Setting 2 in Table 4, the initial conditions at $t = 0$ is $\overline{x} = (0.85; 0.1; 0.05)$. We check the following satisfaction relation:

$$\mathcal{P}_{>0.9}(\text{infected } U^{[0,15]}(\mathcal{P}_{>0.8}\ tt\ U^{[0,0.5]} \text{ infected})).$$

Parameter		Setting 1	Setting 2
Attack	k_1	0.9	5
Inactive computer recovery	k_2	0.1	0.02
Inactive computers getting active	k_3	0.01	0.01
Active computer returns to inactive	k_4	0.3	0.5
Active computer recovery	k_5	0.3	0.5

Table 4: Parameter settings.

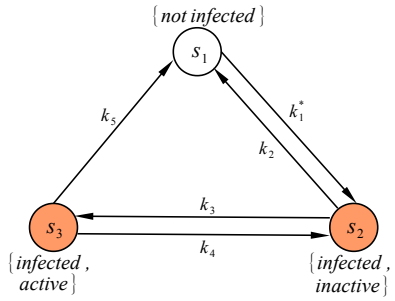

Fig. 7: Example of the CTMC describing computer virus spread.

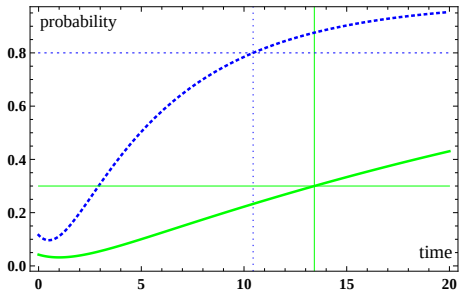

Fig. 8: The green solid line shows $Prob^{\mathcal{M}^l}(s_1, \text{not infected } U^{[0,1]} \text{ infected}, t', t)$. The time-dependent probability $Prob^{\mathcal{M}^l}(s_1, tt \quad U^{[0,0.5]} \text{ infected}, t', t)$ is presented by the blue dotted line.

The formula is split into sub-formulas and the time-dependent satisfaction set of the sub-formula $\Phi_1 = (\mathcal{P}_{>0.8} tt\ U^{[0,0.5]} \text{ infected})$ is calculated first. Similarly to the previous example, the probability $Prob^{\mathcal{M}^l}(s, tt\ U^{[0,0.5]} \text{ infected}, t', t)$ is calculated for all states $s \in S^o$. In Figure 8 this probability at state s_1 is depicted; the probabilities at states s_2 and s_3 equal to one, since these states are already *infected*. We see that the time-dependent satisfaction set is $Sat(\Phi_1, t', t) = \{s_2, s_3\}$ for all $t \in [0, 10.443]$ and $Sat(\Phi_1, t', t) = \{s_1, s_2, s_3\}$ for all $t \in (10.443, 15]$.

The next task is calculating the probability $Prob^{\mathcal{M}^l}(s, \text{infected } U^{[0,15]} \Phi_1, t', t)$. The reachability probability for the time-varying satisfaction set of Φ_1 is calculated following the algorithm mentioned above in this section. We first calculate all discontinuity points $T_0 = 0$, $T_1 = 10.443$ and $T_2 = 15$. An extra state s^* is added and an indicator matrix $\zeta(T_1)$ is constructed: $\zeta(T_1)_{s^*,s^*} = 1$, $\zeta(T_1)_{s_1,s_2} = 0$ for all $s_1, s_2 \neq s^*$. The transient probabilities on time intervals $[0, 10.443)$ and $(10.443, 15]$ are calculated using forward Kolmogorov equation:

$$\Pi'(0, 10.443) = \begin{pmatrix} 0.53 & 0 & 0 & 0.47 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\Pi'(10.443, 15 - 10.443) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Equation (22) is used to calculate $\Upsilon(0, 15)$:

$$\Upsilon(0, 15) = \begin{pmatrix} 0 & 0 & 0 & 0.47 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Equation (23) is used in order to calculate the reachability probability for each state $s \in S^o$: $\pi^{\mathcal{M}^l[\neg \text{infected} \vee \Phi_1]}_{s_1,s^*}(0, 15) = 0.47$; $\pi^{\mathcal{M}^l[\neg \text{infected} \vee \Phi_1]}_{s_2,s^*}(0, 15) = 1$; $\pi^{\mathcal{M}^l[\neg \text{infected} \vee \Phi_1]}_{s_3,s^*}(0, 15) = 1$. The probability $Prob^{\mathcal{M}^l}(s, \text{infected } U^{[0,15]} \Phi_1, t')$ is calculated according to Equation (19), and equals to 0, 1, and 1 for states s_1, s_2, and s_3 respectively. Therefore only states s_2 and s_3 satisfying the formula

$$\mathcal{P}_{>0.9}(\text{infected } U^{[0,15]}(\mathcal{P}_{>0.8}\ tt\ U^{[0,0.5]}\ \text{infected})).$$

In this section we illustrated how the properties of a singe random object in a large communication network (system of interacting objects). Next to the fluid model checking the reader might be interested in the techniques for calculation *fluid passage time*, as discussed in [24] and an MF-CSL logic, which allows checking properties of the overall mean-field model via properties of the individual object [32].

7 Conclusions

In this paper we illustrate several aspects of applying mean-field approximations for efficient analysis of large scale stochastic models of computer systems. Our focus is into providing a self-contained and accessible presentation for beginners.

First, in Sections 2 and 3, we illustrate the basic theory behind mean-field approximation and we describe a systematic approach to applying this technique. Then, in Section 4, we illustrate in full details a non trivial example modeling the dynamics of a bot-net within a computer network. This example shows how to apply the classical results of Section 3 for studying the dynamics of the bot-net for a large number of computers. We discuss the results obtained through several experimental sessions and we have shown the practical efficiency of mean-field approximation. In that section we also illustrate a further application area for mean-field approximation, that is performance and cost evaluation for optimization.

In Section 5 we show a more advanced application of mean-field techniques, where local aspects and inhomogeneity of the systems are taken into account. There, the modeling of spatial aspects is crucial for obtaining a detailed model. We show a possible approach to modeling space, by considering locations and

parameters depending on locations. A further aspect we consider in that section concerns the mean-field approximation of stochastic processes over uncountable domains. This is a rather advanced topic and falls outside the applicability of Theorems 1 and 2. Therefore it requires one to develop ad-hoc results and techniques, following the general idea of mean-field approximation. Despite this complexity, the adoption of uncountable domains can be relevant whenever one is interested into approximating measures that are inherently continuous, such as the aging of certain information, in the considered example.

Finally, in Section 6, we consider a very recent application of mean-field approximation: namely, the use of fast simulation techniques for model checking the behavior of a few stochastic agents within a large scale system. To illustrate the use of this new technique, we consider a concrete example which is a simplified variant of the example considered in Section 4 and we prove some interesting properties, avoiding to fall into the state-space explosion typical of large Markov models.

We believe this paper is a reasonable attempt to give a wide, yet concrete, overview of the main motivations and potentialities of the use of mean-field approximation for modeling and analysis of large scale systems. Mean-field approximation cannot be considered as a ready solution to the state-space explosion problem. Indeed, it is an approximation technique that must be applied carefully [39] and it provides a satisfactory first approximation of a system dynamics which requires, then, to be studied in further details to obtain a more precise analysis, as discussed in Section 1. However, there are already many frameworks that allow for systematic application of mean-field techniques [11,27,42], ensuring a wide reach for the use of these techniques.

References

1. F. Baccelli, F. I. Karpelevich, M. Y. Kelbert, A. A. Puhalskii, A. N. Rybko, and Y. M. Suhov. A mean-field limit for a class of queueing networks. *Journal of Statistical Physics*, 66:803–825, February 1992.
2. C. Baier, B.R. Haverkort, H. Hermanns, and J.P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Trans. Softw. Eng.*, 29(7):524–541, 2003.
3. M. Benaïm and J-Y. Le Boudec. A class of mean field interaction models for computer and communication systems. *Perform. Eval.*, 65(11-12):823–838, 2008.
4. M. Benaïm and J. W. Weibull. Deterministic approximation of stochastic evolution in games. *Econometrica*, 71(3):pp. 873–903, 2003.
5. P. Billingsley. *Probability and Measure*. Wiley-Interscience, 3 edition, 1995.
6. A. Bobbio, M. Gribaudo, and M. Telek. Analysis of large scale interacting systems by mean field method. In *Quantitative Evaluation of Systems, 2008. QEST '08. Fifth International Conference on*, pages 215–224, 2008.
7. L. Bortolussi. Hybrid limits of continuous time markov chains. In *QEST*, pages 3–12. IEEE Computer Society, 2011.
8. L. Bortolussi and J. Hillston. Fluid model checking. In M. Koutny and I. Ulidowski, editors, *CONCUR*, volume 7454 of *Lecture Notes in Computer Science*, pages 333–347. Springer, 2012.

9. L. Bortolussi and J. Hillston. Fluid model checking. In *CONCUR*, volume 7454 of *LNCS*, pages 333–347. Springer, 2012.
10. L. Bortolussi, J. Hillston, D. Latella, and M. Massink. Continuous approximation of collective systems behavior: a tutorial. Technical Report cnr.isti/2011-TR-021, ISTI CNR, 2011.
11. L. Bortolussi, J. Hillston, D. Latella, and M. Massink. Continuous approximation of collective systems behaviour: A tutorial. *Performance Evaluation*, (0):–, 2013.
12. Luca Bortolussi and Jane Hillston. Fluid model checking. *CoRR*, abs/1203.0920, 2012.
13. Cabspotting. http://stamen.com/clients/cabspotting.
14. A. Chaintreau, J. Le Boudec, and N. Ristanovic. The age of gossip: spatial mean field regime. In *Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, SIGMETRICS '09, pages 109–120, New York, NY, USA, 2009. ACM.
15. A. Chaintreau, J-Y. Le Boudec, and N. Ristanovic. The age of gossip: spatial mean field regime. In J. R. Douceur, A. G. Greenberg, T. Bonald, and J. Nieh, editors, *SIGMETRICS/Performance*, pages 109–120. ACM, 2009.
16. F. Ciocchetta and J. Hillston. Bio-pepa: A framework for the modelling and analysis of biological systems. *Theor. Comput. Sci.*, 410(33-34):3065–3084, 2009.
17. R. W. R. Darling. Fluid Limits of Pure Jump Markov Processes: a Practical Guide. *ArXiv Mathematics e-prints*, October 2002.
18. R.W.R. Darling and J.R. Norris. Differential equation approximations for markov chains. *Probability Surveys*, 5:37–79, 2008.
19. D. D. Deavours, G. Clark, T. Courtney, D. Daly, S. Derisavi, J. M. Doyle, W. H. Sanders, and P. G. Webster. The Mobius framework and its implementation. *Software Engineering, IEEE Transactions on*, 28(10):956–969, 2002.
20. N. Gast and B. Gaujal. A mean field model of work stealing in large-scale systems. In V. Misra, P. Barford, and M. S. Squillante, editors, *SIGMETRICS*, pages 13–24. ACM, 2010.
21. C. S. Gillespie. Moment closure approximations for mass-action models. *IET Systems Biology*, 3:52–58, 2009.
22. D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81(25):2340–2361, December 1977.
23. R. 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 2010)*, August 2010.
24. R. Hayden, A. Stefanek, and J. T. Bradley. Fluid computation of passage time distributions in large Markov models. *Theoretical Computer Science*, 413(1):106–141, January 2012.
25. R. A. Hayden and J. T. Bradley. A fluid analysis framework for a markovian process algebra. *Theor. Comput. Sci.*, 411(22-24):2260–2297, 2010.
26. J. Hillston. *A compositional approach to performance modelling*. Cambridge University Press, New York, NY, USA, 1996.
27. J. Hillston. Fluid flow approximation of pepa models. In *QEST*, pages 33–43. IEEE Computer Society, 2005.
28. J. Hillston, M. Tribastone, and S. Gilmore. Stochastic process algebras: From individuals to populations. *The Computer Journal*, 2011.
29. L. P. Kadanoff. More is the Same; Phase Transitions and Mean Field Theories. *Journal of Statistical Physics*, 137:777–797, December 2009.

30. A. Kleczkowski and B. T. Grenfell. Mean-field-type equations for spread of epidemics: the small world model. *Physica A: Statistical Mechanics and its Applications*, 274(12):355 – 360, 1999.
31. A. Kolesnichenko, A. Remke, P.-T. de Boer, and B.R. Haverkort. Comparison of the mean-field approach and simulation in a peer-to-peer botnet case study. In *8th European Performance Engineering Workshop (EPEW'11)*, volume 6977 of *Lecture Notes in Computer Science*, pages 133–147. Springer, 2011.
32. A. Kolesnichenko, A. Remke, P.T. de Boer, and B.R. Haverkort. A logic for model-checking mean-field models. Technical report, University of Twente, 2013. Accepted for publication in PDS.
33. T.G. Kurtz. Solutions of ordinary differential equations as limits of pure jump markov processes. *Journal of Applied Probability*, 7(1):49–58, 1970.
34. T.G. Kurtz. *Approximation of population processes*, volume 36. Society for Industrial Mathematics, 1981.
35. M. Z. Kwiatkowska, G. Norman, and D. Parker. Stochastic model checking. In M. Bernardo and J. Hillston, editors, *SFM*, volume 4486 of *Lecture Notes in Computer Science*, pages 220–270. Springer, 2007.
36. J-Y. Le Boudec, D. McDonald, and J. Mundinger. A generic mean field convergence result for systems of interacting objects. In *Proceedings of the Fourth International Conference on Quantitative Evaluation of Systems*, QEST '07, pages 3–18, Washington, DC, USA, 2007. IEEE Computer Society.
37. W.D. McComb. *Renormalization Methods: A Guide For Beginners*. OUP Oxford, 2004.
38. M. Mitzenmacher. The power of two choices in randomized load balancing. *IEEE Trans. Parallel Distrib. Syst.*, 12(10):1094–1104, October 2001.
39. A. Pourranjbar, J. Hillston, and L. Bortolussi. Dont just go with the flow: Cautionary tales of fluid flow approximation. In M. Tribastone and S. Gilmore, editors, *Computer Performance Engineering*, volume 7587 of *Lecture Notes in Computer Science*, pages 156–171. Springer Berlin Heidelberg, 2013.
40. 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.
41. M. Tribastone. Relating layered queueing networks and process algebra models. In A. Adamson, A. B. Bondi, C. Juiz, and M. S. Squillante, editors, *WOSP/SIPEW*, pages 183–194. ACM, 2010.
42. M. Tribastone, S. Gilmore, and J. Hillston. Scalable differential analysis of process algebra models. *IEEE Trans. Software Eng.*, 38(1):205–219, 2012.
43. N. G. Van Kampen. *Stochastic Processes in Physics and Chemistry*. North-Holland Personal Library. Elsevier Science, 2011.
44. E. van Ruitenbeek and W. H. Sanders. Modeling peer-to-peer botnets. In *5th Int. Conference on Quantitative Evaluation of SysTems, (QEST'08)*, pages 307–316. IEEE CS Press, 2008.
45. Wolfram Research, Inc. Mathematica tutorial. `http://reference.wolfram.com/mathematica/tutorial/IntroductionToManipulate.html`, 2010.

INSTITUTE FOR ADVANCED STUDIES LUCCA