Applying Mean-field Approximation to Continuous Time Markov Chains

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

[1] DACS, University of Twente, The Netherlands
{a.v.kolesnichenko, a.k.i.remke}@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 system 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 agent 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. 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 existing model checking analysis techniques.

1 Introduction

Mean Field Approximation originated in statistical physics [1] and is a technique developed within the field of probability theory. This technique is useful

This is the authors post-print version of the paper published in special issue of Lecture Notes in Computer Science Volume 8453. The final publication is available at www.springerlink.com. Citations to this paper should be as follows: Kolesnichenko, A.V. and Senni, V. and Pourranjabar, A. and Remke, A.K.I. Applying Mean-Field Approximation to Continuous Time Markov Chains. In: Stochastic Model Checking. Rigorous Dependability Analysis Using Model Checking Techniques for Stochastic Systems. Lecture Notes in Computer Science 8453. Springer Verlag, pp. 242-280. 2014. ISSN 0302-9743 ISBN 978-3-662-45488-6

to study the behaviour 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 behaviour 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 [23,31]. Beyond physics, this approximation technique is applied in studies of epidemics models [24], queueing theory [6,1], and network performance [30,11].

In this tutorial, the stochastic systems we are interested in typically consist of a relatively small number of particle types. The particles of each type often have a simple behaviour and are replicated many times to form large populations. Their interaction may give rise to a complex behaviour and patterns that can not be found considering the single particle, but emerge by their interaction. Mean-field approximation is used to model and analyse efficiently the so-called *emergent behaviour* of such large-scale systems. Classical applications of this technique generally require two abstractions. The first is that when studying the system, one abstracts away from the particles' identities, and instead of capturing the behaviour of each instance, the system's behaviour is observed at the level of populations [22]. The second abstraction suggests that the spatial distribution of the agents across the system locations is ignored, 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 [17,37]). In this tutorial we illustrate both a classical application (Section 3) and a more sophisticated modelling where space inhomogeneity has a significant impact on the system's emergent behaviour (Section 4).

The core idea of the mean-field method is to approximate the dynamics of a Markov population process through a system of differential equations [27]. The result is a reliable approximation when the population size is sufficiently large, since under specific conditions the behaviour of the system tends to the deterministic dynamics captured by the differential equations. In this case, one additional important property is the *decoupling assumption*; that is the joint probability distribution associated with the system can be expressed as the product of the marginals. This property allows to study the behaviour of individual particles within the whole system in an efficient way.

A closely related approximation technique is known as *moment closure* [16]. This technique allows to estimate the first few moments of a stochastic process by a closed system of equations. Mean-field approximation can be seen as a form of moment closure where the second moment (variance) and the higher moments have been set to zero. The first-order approximation is often very coarse and can potentially lead to misleading results [33]. In practice, however, it can be used to gain some insights about the average or the global behaviour of the system at a relatively low cost.

When first-order or mean-field approximation is applied, the resulting model can be described in terms of a deterministic system, as mentioned previously. In the literature this is often referred to as *deterministic approximation* [4,9].

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 [37].

Continuous Time Markov Chains are often used to provide a stochastic semantics to process algebra used in performance modelling of computer systems [20]. However, stochastic process algebra models of realistic size can easily result in very large and intractable state spaces. In that context a technique called *fluid-flow approximation* [21] has been used to construct a continuous state-space representation of the underlying discrete state-space, and ordinary differential equations are used to describe their dynamics. This technique is justified by results on mean-field approximation of Continuous Time Markov Chains [36,22,19]. 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 [34].

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 [9], 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 [13].

2 Preliminaries

In this paper we consider systems consisting of large populations of interacting objects. Such systems are common in biology and chemistry, as well as in telecommunications and queueing theory [3,12,22,35]. Due to the problem of state space explosion, the models of such systems are often unmanageable for the purpose of analysis and are not suitable for direct application of classic analysis techniques such as simulation and model checking. In this tutorial we address the modelling and analysis of such models using *mean-field method*.

The main idea of the mean-field analysis is to describe the evolution of a population that is composed of many similar objects via a deterministic behaviour. It states that under certain assumptions on the dynamics of the system and when the size of the population grows, the ratio of the system's *variance* to the size of the state space tends to zero. Therefore, when the population is large, the stochastic behaviour of the system can be studied through the *unique solution* of a system of Ordinary Differential Equations (ODE) defined by using the limit dynamics of the whole system.

Since the purpose of this tutorial is to provide the guided examples of the application of the mean-field method, we will not be discussing the detailed theoretical background of the mean-field method (see, e.g. [9]). Instead, we present the modelling procedure from the practical point of view. We build the model of the whole population based on the behaviour of the random individual object.

2.1 Model definition

Let us start with a random individual object in the large population. We assume that the size of the population N is constant and do not distinguish between the classes of the individual objects for the simplicity of the notation. However, this assumptions can be relaxed, see, e.g., Section 4 of the current tutorial.

The behaviour of such an object can be described by defining the states or "modes" this object experiences during its lifetime, and the transitions between these states. Formally, the individual or local model (the model of the random object in the population) is defined as follows:

Definition 1 (Local model). *A local model $\mathcal{X}$ describing the behaviour of one object is constructed as a tuple $(S, \mathbf{Q}, L)$ that consists of a finite set of K local states $S = \{s_1, s_2, ..., s_K\}$; the infinitesimal generator matrix $\mathbf{Q}$ which may depend on the overall system state; and the labelling function $L : S \rightarrow 2^{LAP}$ that assigns local atomic propositions from a fixed finite set of Local Atomic Properties (LAP) to each state.* □

Self-loops are assumed to be eliminated. The generator matrix $\mathbf{Q}$ is a matrix $S \times S$, whose entries describe the rate at which an individual object changes states. The $\mathbf{Q}$ may potentially depend on the system's overall state. We discuss the transitions rates of the individual objects later in this section.

Given the large number N of objects, we build the overall model of the whole population. Instead of modelling each object individually, which would lead to the state-space explosion problem, we (i) lump the state space; (ii) normalize the population, and (iii) check whether the convergence of the behaviour to the deterministic limit holds and build *the overall mean-field model* X, using the local model $\mathcal{X}$. Let us first provide the explanations on the way this model is built, which will be followed by the definition of the overall (or global) model.

If the identity of each object is preserved, the state space of the model of the whole population $\mathcal{X}^{(N)}$ will potentially consists of K^N states, where K is the number of states of the local model. However, due to the identical and unsynchronized behaviour of the individual objects the *counting abstraction* is applied to find the stochastic process X, whose states capture the distribution of the individual objects across the states of the local model $\mathcal{X}$. In general, the transition rates may depend on the state of the overall model, $\overline{X}(t)$. Therefore, using the counting abstraction the generator matrix $\mathbf{Q}(\overline{X}(t))$ is constructed as in [6]:

$$\mathbf{Q}_{i,j}(\overline{X}(t)) = \begin{cases} \lim_{\mathit{\Delta} \to 0} \frac{1}{\mathit{\Delta}} \text{Prob}(\mathcal{X}(t+\mathit{\Delta})) = j | \mathcal{X}(t) = i, \overline{X}(t)), & \text{if } \ X_i(t) > 0, \\ 0, & \text{if } \ X_i(t) = 0, \\ -\sum_{h \in S, j \neq i} \mathbf{Q}_{i,h}(\overline{X}(t)), & \text{for } i = j, \end{cases}$$

where $\mathcal{X}(t)$ is a state of the local model at time t.

The first step for the construction of the mean field model is to normalize the state vector. The normalized state space is as follows: $\overline{x}(t) = \overline{X}(t)/N$, where $0 \leq \overline{x}_i(t) \leq 1$; and the related transition rates are $Q_{i,j}^{(N)}(\overline{x}(t)) = \mathbf{Q}_{i,j}(N \cdot \overline{x}(t))$.

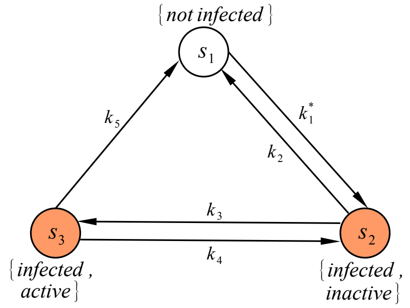

Fig. 1: The model describing computer virus spread.

In this tutorial we only consider models which satisfy a condition known as *density dependence*. This condition requires that there exists a matrix of rate functions that is constant for all the normalised models in a sequence of models with increasing sizes. This means that transition rates scale together with the model population, so that in the normalized models they are independent of the population. Formally, in the limit of $N \to \infty$, the matrix of rate functions (generator matrix $Q_{i,j}(\overline{x}(t))$) satisfies $Q_{i,j}(\overline{x}(t)) = Q^{(N)}_{i,j}(\overline{x}(t))$ for all $N > 1$.

The existence and properties of $Q_{i,j}(\overline{x}(t))$ play a crucial role in the applicability of the mean-field theory to the given sequence of local models and building the overall model. In the context of the models which satisfy density dependence, the rate functions are required to be Lipschitz-continuous. Secondly, the model should satisfy *convergence of the initial occupancy vector*. The limit theorem which relies on these assumptions will be covered later. First, let us state the construction of the mean-field model.

Definition 2 (Overall mean-field model). *An overall mean-field model X describes the limit behaviour of $N \to \infty$ identical objects, each modelled by $\mathcal{X}$, and is defined as a tuple (X, Q), that consists of an infinite set of states*

$$X = \{\overline{x} = (x_1, x_2, \ldots, x_K) | (\forall j \in \{1, \ldots, K\}, x_j \in [0,1] \wedge \sum_{i=1}^{K} x_i = 1)\},$$

where $\overline{x}$ is called occupancy vector, and $\overline{x}(t)$ is the value of the occupancy vector at time t; x_j denotes the fraction of the individual objects that are in state s_j of the local model $\mathcal{X}$. The transition rate matrix $Q(\overline{x}(t))$ consists of entries $Q_{s,s'}(\overline{x}(t))$ that describe the transition of the system from state s to state s'. □

Example 1. In the following we describe a simple model of the virus spread in the population of interacting computers of size N. We start with the local model (see Figure 1). The states of $\mathcal{X}$ represent the 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 $S = \{s_1, s_2, s_3\}$ with $|S| = K = 3$ states. They are labelled as *infected*, *not infected*, *active* and *inactive*, as indicated in Figure 1.

Given a system of N such computers, we can model the limiting behaviour of the whole system through the overall mean-field model, which has the same underlying structure as the individual model (see Figure 1), however, with state space $\overline{x} = \{x_1, x_2, x_3\}$, where x_1 denotes the fraction of not-infected computers, and x_2 and x_3 denote the fraction of active and inactive infected computers, respectively. For example, a system without infected computers is in state $\overline{x} = (1, 0, 0)$; a system with 50% not infected computers and 40% and 10% of inactive and active infected computers, respectively, is in state $\overline{x} = (0.5, 0.4, 0.1)$.

The transition rates $k_1^*,\ k_2,\ k_3,\ k_4,\ k_5$ represent the following: the infection rate k_1^*, the recovery rate for an inactive infected computer k_2, the recovery rate for an active infected computer k_5, and the rates with which computers become active k_3 and return to the inactive state k_4. Rates $k_2, k_3, k_4, and\ k_5$ are specified by the individual computer and computer virus properties and do not depend on the overall system state. The infection rate k_1^* does depend on the fraction of computers that is infected and active and the fraction of not-infected computers. We discuss the generator matrix in the next example.

2.2 Mean-field analysis

We stated $\mathcal{X}$ represents the behaviour of each object and X represents the limiting behaviour of N identical objects. The model respects the density dependence condition. Here we express a reformulation of the Kurtz's theorem which relates the behaviour of the sequence of models with increasing sizes to the limit behaviour. Assuming that functions in $Q_{i,j}(\overline{x}(t))$ are Lipschitz-continuous and for increasing values of the system size, the initial occupancy vectors converge to $\overline{x}(0)$, then when $N \to \infty$, the sequence of local models *converges almost surely* [5] to the occupancy vector $\overline{x}$.

Theorem 1 (Mean-field convergence theorem). *The normalized occupancy vector $\overline{x}(t)$ at time $t < \infty$ tends to be deterministic in distribution and satisfies the following differential equations when N tends to infinity:*

$$\frac{d\overline{x}(t)}{dt} = \overline{x}(t) \cdot Q(\overline{x}(t)),\ \ given\ \overline{x}(0). \tag{1}$$

□

The ODE (1) is called *limit ODE.* It provides the results for $N \to \infty$, which is not the case for a real-life models. When the number of objects in the population is finite, but sufficiently large the limit ODE provides an accurate approximation of the mean of the occupancy vector $\overline{x}(t)$ over time.

The transient analysis of the overall system behaviour can be performed using the above system of differential equations (1), i.e., the fraction of the objects in each state of $\mathcal{X}$ at every time t is calculated, starting from some given initial occupancy vector $\overline{x}(0)$.

For models considered in practice, however, the assumption of density dependence may be too restrictive [13]. Furthermore, also the assumption of (global) Lipschitz continuity of transition rates can be unrealistic [7]. Therefore, this assumptions can be relaxed and 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, can be obtained. We will not be focusing on the reformulation of the convergence theorem here, instead we refer to [9], and provide the following example.

Example 2. In the following we provide an example of applying the mean-field method to the virus spread model, as in Example 1. We explain how to obtain the ODEs, describing the behaviour of the system and produce performance evaluation measures.

As was discussed in the previous example, all transition rates of a single computer model are constant, but k_1^*. This rate depends on how often a not infected computer gets attacked. In this example we assume that the virus is "smart enough" to attack not infected computers only. The infection rate then might be seen as the number of attacks performed by all active infected computers, which is distributed over all not-infected computes in a chosen group:

$$k_1^*(\overline{x}(t)) = k_1 \cdot \frac{x_3(t)}{x_1(t)},$$

where $\overline{x}(t) = (x_1(t), x_2(t), x_3(t))$ represents the fraction of computers in each state at time t, and k_1 is the attack rate of a single active infected computer.

The transition rates are collected to the generator matrix:

$$Q(\overline{X}(t)) = \begin{pmatrix} -k_1^*(\overline{x}(t)) & k_1^*(\overline{x}(t)) & 0 \\ k_2 & -(k_2+k_3) & k_3 \\ k_5 & k_4 & -(k_4+k_5) \end{pmatrix} \tag{2}$$

Then Theorem 1 is used to derive the system of ODEs (1), that describes the mean-field model:

$$\begin{cases} \dot{x}_1(t) = -k_1 x_3(t) + k_2 x_2(t) + k_5 x_3(t), \\ \dot{x}_2(t) = (k_1 + k_4) x_3(t) - (k_2 + k_3) x_2(t), \\ \dot{x}_3(t) = k_3 x_2(t) - (k_4 + k_5) x_3(t). \end{cases} \tag{3}$$

To obtain the distribution of the objects between the states of the model over time the above ODEs have to be solved.

The convergence theorem does not explicitly cover the asymptotic behaviour (i.e. limit in time). However, when certain assumptions hold, the mean-field equations allow to perform various studies including steady state analysis of the population models as well as model checking [8]. We will not cover the details here and the interested reader is referred to [3]. We will use mean-field for steady state analysis in Section 4.

3 Mean-field Analysis of a Botnet

In this section we discuss the applicability of the mean-field method to modelling peer-to-peer botnet, as in [26] . In Section 3.1 we discuss the characteristics of the botnet, which are important for modelling. Section 3.2 describes the mean-field model of the botnet spread. The performance evaluation results are presented in Section 3.3, together with an example of wider usability of the mean-field model.

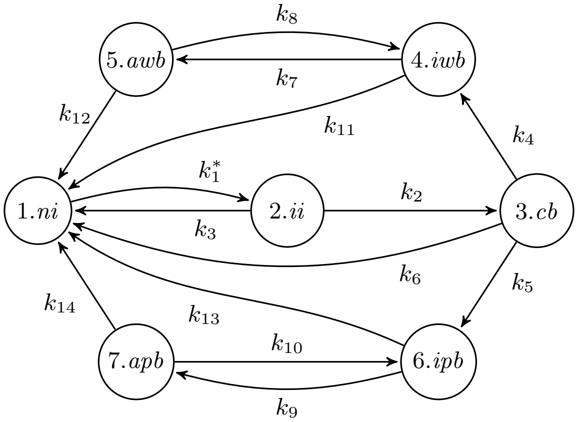

Fig. 2: Possible states of a computer in the network. The shorthand names are defined as follows: *ni*=*NotInfected*, *ii*=*InitialInfection*, *cb*=*ConnectedBot*, *iwb*=*InactiveWorkingBot*, *awb*=*ActiveWorkingBot*, *ipb*=*InactivePropagationBot*, and *apb*=*ActivePropagationBot*.

3.1 Description of the System

Let us describe the steps each computer goes through during the botnet spread. These are similar to the examples in the previous section, however, the current Botnet model is more detailed (see Figure 2) and comply the realistic botnet behaviour.

The computer which is in the *NotInfected* state (S_1) enters the *InitialInfection* (S_2) state with rate k_1^*. Then, it attempts to connect to the other bots in the botnet; if the connection is successful the computer goes tot he *ConnectedBot* state (S_3) with rate k_2. The initially infected computer recovers and returns to the state S_1 with rate k_3. After connecting to the botnet, computer downloads a malware and joins the botnet either as *InactiveWorkingBot* (S_4) or as *InactivePropagationBot* (S_6) with rates k_4 and k_5, respectively; otherwise, the computer recovers from the connected state with the 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_{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 recovery rates from *InactiveWorkingBot* and *ActiveWorkingBot* are k_{11}, k_{12}, respectively.

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.

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 *Working* 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 $k_2 \ldots k_{14}$ are constant for each computer, while rate k_1^* to move from the *Not-Infected* 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.

3.2 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 $\overline{X} = \langle X_1, X_2, \ldots X_7 \rangle$ counts how many computers are in states $S_1, ..., S_7$. The occupancy measure is found by normalizing $\overline{\mathbf{X}}$ into $\overline{x}$.

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 $\overline{x}(t)$ of the system. The rate matrix $R(\overline{x}(t))$ of the model is given as:

$$R(\overline{x}(t)) = \begin{pmatrix} 0 & k_1^* & 0 & 0 & 0 & 0 & 0 \\ k_3 & 0 & k_2 & 0 & 0 & 0 & 0 \\ k_6 & 0 & 0 & k_4 & 0 & k_5 & 0 \\ k_{11} & 0 & 0 & 0 & k_7 & 0 & 0 \\ k_{12} & 0 & 0 & k_8 & 0 & 0 & 0 \\ k_{13} & 0 & 0 & 0 & 0 & 0 & k_9 \\ k_{14} & 0 & 0 & 0 & 0 & k_{10} & 0 \end{pmatrix} \tag{4}$$

The $|S| \times |S|$ infinitesimal generator matrix $Q(\overline{x}(t))$ is given as follows: Q_{s_1,s_2} is equal to the transition rate R_{s_1,s_2} to move from the state s_1 to the state s_2 and $Q_{s,s}$ is equal to the negative the sum of all the rates in row s. In a given example the only rate which depends on a state of the system is the infection rate $k_1^*(\overline{x}(t))$, 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 $k_1 \cdot x_7(t)$. These infections are spread out randomly over all not-yet infected computers, whose number is denoted by $x_1(t)$. Hence, the infection rate k_1^* perceived by each individual computer is given by the ratio:

$$k_1^*(\overline{x}(t)) = \frac{k_1 \cdot x_7(t)}{x_1(t)}. \tag{5}$$

Once 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. Therefore, the initial value problem we study is defined as follows:

$$\frac{d\,\overline{x}(t)}{dt} = \overline{x}(t) Q(\overline{x}(t)), \qquad \text{with initial condition } \overline{x}(0). \tag{6}$$

The system of equations we obtain is:

$$\begin{cases} \dot{x}_1(t) &= k_3x_2(t) + k_6x_3(t) + k_{11}x_4(t) \\ & \quad +k_{12}x_5(t) + k_{13}x_6(t) + (k_{14} - k_1)x_7(t) \\ \dot{x}_2(t) &= -(k_2 + k_3)x_2(t) + k_1x_7(t) \\ \dot{x}_3(t) &= k_2x_2(t) - (k_4 + k_5 + k_6)x_3(t) \\ \dot{x}_4(t) &= k_4x_3(t) - (k_7 + k_{11})x_4(t) + k_8x_5(t) \\ \dot{x}_5(t) &= k_7x_4(t) - (k_8 + k_{12})x_5(t) \\ \dot{x}_6(t) &= k_5x_3(t) - (k_9 + k_{13})x_6(t) + k_{10}x_7(t) \\ \dot{x}_7(t) &= k_9x_6(t) - (k_{10} + k_{14})x_7(t) \end{cases} \tag{7}$$

The equations can be solved analytically, however the closed forms are impractically large. We used Wolfram Mathematica [39] to obtain the analytical solution.

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.

3.3 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 behaviours, and their impact on the system behaviour.

The simulation of the model was done using the Möbius tool [14] as in [38]. Each experiment covered one week of simulated time; it was replicated 1000 times; the mean values and 95% confidence intervals of the measures of interest are obtained. The initial conditions for each experiment are as follows: 200 computers are located in the place *ActivePropagationBots*.

We use Wolfram Mathematica [39] to obtain solutions for the set of differential equations (7) 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.

We first consider Baseline experiment. 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 experiment together with the 95% confidence intervals of the Möbius 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.

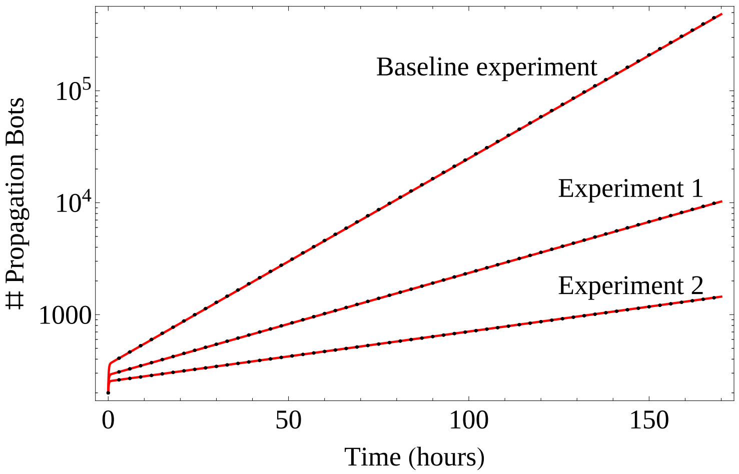

Fig. 3: Number of propagation bots over time in the Baseline experiment and experiments 1 ad 2 obtained from mean-field approximation together with the confidence intervals (black bars) obtained from the simulation.

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.

To investigate how a reduced infection spread would influence the growth of botnets, Experiments 1 and 2 were done in [38]. 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 [38] and [26] 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 behaviour 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 anti-virus software developers to find the better strategy for botnet removal.

Next, we introduce a cost concept to analyse 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 anti-virus 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 anti-virus software becomes very expensive. We analyse this trade-off in more detail in the following. We calculate the cumulative cost between t_0 and t_1 as follows:

$$C(t_0, t_1, RR, D_1, D_2) \;=\; \int_{t_0}^{t_1} \left(D_1 \cdot \mathrm{IC}(t, RR) + D_2 \cdot RR \cdot \mathrm{AC} \right) dt \qquad (8)$$

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 (D_1); 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 $[0.001, 10]$ we calculate the cost as a function of time for given D_1 and D_2. Results are depicted in Figure 5. 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 anti-malware 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 though not all parameters might be known in reality, such analysis can help to obtain a better understanding of the characteristics of botnet spread.

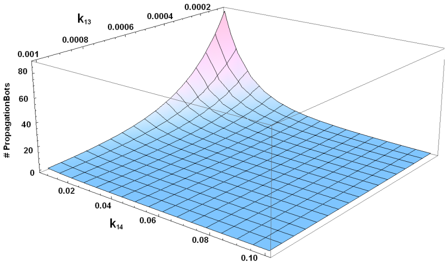

Fig. 4: Number of propagation bots for $(k_{13}, k_{14}) \in [8 \cdot 10^{-5}; 10^{-3}] \times [8 \cdot 10^{-3}; 10^{-1}]$ at time $T = 3 days$, all other parameters are the same as for baseline experiment (see Table 2).

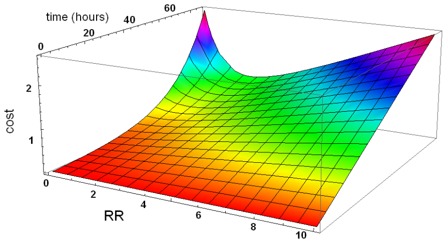

Fig. 5: Cost of the system performance for $D_1 = 0.01, D_2 = 4 \cdot 10^{-5}$.

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.

4 Spatial Mean-Field Models

The mean-field analysis was firstly used in the fields of physics (when studying gas dynamics) and systems biology (studying how concentrations of reactants behave in a solution). In those domains, the assumption is made that the spatial distribution of particles/molecules across the system is homogeneous and the interacting entities are spread across the space uniformly. Such systems are often referred to as *spatially homogeneous*, in physics, and *well-stirred*, in chemistry. When analysing them, regardless of their spatial structure a single rate is assigned for each type of particle-to-particle interaction and these interactions respectively have the same probability to take place at different locations. Therefore, the effect the locations may potentially have on the overall dynamics is abstracted away.

In this section we focus on the appropriateness of the abstraction with respect to the spacial aspects in the context of modelling computer and communication networks. Indeed, depending on the system under study abstracting from the space might be a suitable simplifying step. For example, in the previous section the state vector only counted how many computers are in different local states, regardless of their locations across the geographical space (as a result, the transition rate functions did not depend on the computers' locations). Although this abstraction is reasonable in certain systems, but there exist those whose dynamics and emergent behaviours are significantly dependent on the locations of the constituent interacting objects. For those systems, the model should take into account the spatial aspects (the location of the entities, their distance, etc.) or else, the system's behaviour may not be captured effectively.

In this section, we consider an example of a large-scale peer-to-peer gossip network [11] where the emergent behaviour of the system significantly depends on locations of the objects involved. We describe how the mean-field equations are constructed in a way that the effect the locations have on the system's behaviour is also captured.

An additional feature of the example we review in this section is that it shows a case where the mean-field method is applied to a *uncountable* space. In Section 3, the method was applied to a finite-domain CTMC. Nevertheless, Kurtz's Theorem [27] has the potential to be applied also to Markov chains defined over uncountable domains [32]. As we will express, in the model we consider some of the state variables range over positive real numbers and this complicates the process of applying the method as the mean-field equations consists of partial differential equations. Here, we will review how the mean-field equations are practically constructed and avoid the proof of convergence. The more interested reader can refer to [11] for that purpose.

4.1 The Age of Gossip

We consider the example in [11], a model proposed for a peer-to-peer opportunistic communication network. Two types of entities are present in this network: some are mobile agents and can move through different locations, and some others are the stationary base stations. The base stations transmit fresh updates on a piece of data by the wireless medium and these updates are received by the mobile agents when they are close to one of the base stations. The data the base stations send is time-stamped. The age of a piece of data an agent holds is defined to be the time elapsed since it was transmitted by one of the base stations. Therefore, the age of data *just* received is zero. The age of an agent is defined to be the age of the data it holds. In addition to the data exchanges with the base stations, the mobile agents are capable of radio communication between themselves. If two such agents are close enough, the one who has the most recent version transmits its data to the other. This mechanism helps the agents receive updated data even if they have not directly visited a base station.

The system consists of a number of *locations* through which the mobile agents move. We assume that the base stations in each location 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 they both belong to the same location or when they are in two different locations. The latter captures the situation when agents are close to the borders of their location and can potentially exchange data with agents of the other locations.

Formal Model Description. Let $L = \{1, 2, \ldots, C\}$ be the set of locations and N denote the number of mobile agents. For the i^{th} agent, we define $X_i \in \mathbb{R}^+$ to denote its age and $c_i \in L$ to represent its location. Hence, the state vector is $\boldsymbol{\xi} = \langle X_1, X_2, \ldots X_N, c_1, c_2 \ldots c_N \rangle$. Now we define the transitions which affect the system's state and the rate functions associated with these transitions.

1. **Mobility.** An agent moves from location c to c' with rate $\rho_{c,c'}, c \neq c'$. If there are N_c agents in c, the total rate at which agents from c move to c' is $N_c \times \rho_{c,c'}$.
2. **Contact with base station**. An agent i with age X_i in $c \in L$ may contact a base station in c and get fresh data. As the result, $X_i = 0$. For each location c a parameter μ_c describes the rate at which an agent in c receive data directly from base stations in c. If no base stations are in c, then $\mu_c = 0$.
3. **Opportunistic contact within locations.** An agent i in a location c opportunistically communicates with any of the other $N-1$ agents with rate $2\eta_c/(N-1)$. The total rate of communications observed between mobile agents in c is determined by two factors: the number of agents the location contains and its topological structure. The larger the number of agents is, the higher the frequency of the communication. However, when two locations have exactly the same number of agents, the respective rates of the meetings may not be the same, as the structural properties of one might encourage agent-to-agent interaction more than the other. Hence, for each $c \in L$, a parameter η_c is defined, which captures how effectively the location's structure encourages the such interactions. If there are N_c agents in location c, the total rate at which agents communicate between themselves is:

$$\binom{N_c}{2} \times \frac{2\eta_c}{(N-1)} = \frac{(N_c) \times (N_c - 1)}{N-1}\eta_c. \tag{9}$$

4. **Opportunistic contact across locations.** A mobile agent in a location c may communicate with a mobile agent from a different location c'. This interaction happens with rate $2\beta_{c,c'}/(N-1)$. For each c and c', $(c \neq c')$, $\beta_{c,c'}$ is a constant which affects the rate at which the agents in c communicate with the agents in c'.

The ages of the agents continuously grow unless they communicate with one of the base stations or receive fresher data from other mobile agents. At any point of time and for each location, one can derive the age distribution for the agents in that location. The aim is to construct the network in a way that an acceptable distribution of ages is maintained across all locations.

State Space Representation. The state vector used for capturing the state of a system depends on the system under study and the modelling goals. In the peer-to-peer network we consider, the age of the agents is one of their key properties. Therefore, let the configuration of the system at any time t be captured by a *continuous* distribution $\boldsymbol{\xi}''(z,t)$, $z \in \mathbb{R}^+$, where $\xi''(j,t)$ denotes how many agents have age j at time t. Using this state representation, a *partial differential equations* over the dimensions z and t is formed to effectively study how the age distribution of the agents evolves. However, the modelling suffers from the fact that the mobility of the agents is abstracted away and the effect their locations potentially have on the system's emergent behaviour is not realised. The dynamics of the system is faithfully captured if the state vector takes into account both properties of the agents, i.e. their age and their locations.

Consider $c \in L$. For the i^{th} agent with age X_i, we define the distribution δ_{X_i}, a Dirac mass at X_i. At a time t, the age distribution of agents in c across $\mathbb{R}^+$ is denoted by distribution $M_c^N(t)=\sum_{i=1}^{N} 1_{\{c_i=c\}}\delta_{X_i^N(t)}$, which is a continuous distribution denoting the number of agents who have any age z at location c at time t. The *vector of such distributions* $\boldsymbol{M}^N(t)=\langle\, M_1^N(\cdot,t), M_2^N(\cdot,t),\dots,M_C^N(\cdot,t)\,\rangle$ is capable of capturing both the locations and ages of the agents, and is used in the rest of this section for state state representation of the mean-field analysis.

4.2 Mean-Field Limit Behaviour

In order to derive the deterministic limit behaviour, first we focus on the mobility of the agents across locations and then we consider message propagation.

Mobility of Agents. Let $\boldsymbol{U}^N(t) = \langle\, U_1^N(t), U_2^N(t),\dots,U_C^N(t)\,\rangle$ capture the number of agents in different locations at time t, assuming that there are N agents in the system. Thus, the *location occupancy measure* is defined as: $\bar{\mathbf{U}}^N(t) = \frac{\boldsymbol{U}^N(t)}{N} = \langle \bar{\mathrm{U}}_0^N(t), \bar{\mathrm{U}}_1^N(t),\dots,\bar{\mathrm{U}}_C^N(t)\rangle$ where each $U_c^N(t)_{c\in L}$ represents the fraction of the agents which are in location c at time t. Assume that, when $N\to\infty$, the sequence $\bar{\mathrm{U}}_c^N(0)$ converges to a unique limit:

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

Since the convergence of initial occupancy measure holds and the system satisfies density dependence (rate functions in the normalised system is independent of N), we use Kurtz's Theorem [28] to prove that, *at any time point* $t{>}0$, if $N{\to}\infty$, then process $\bar{\mathbf{U}}^N(t)$ converges to a deterministic limit $\bar{\mathbf{u}}(t) = \langle\bar{\mathrm{u}}_1(t),\bar{\mathrm{u}}_2(t),\dots\bar{\mathrm{u}}_C(t)\rangle$, where $\bar{\mathrm{u}}_c(t)$ is the solution of the following initial value problem:

$$\forall c\in L,\ \ \frac{\partial\ \bar{\mathrm{u}}_c(t)}{\partial t}=\left(\sum_{c'\neq c}\rho_{c',c}\bar{\mathrm{u}}_{c'}\right)-\left(\sum_{c'\neq c}\rho_{c,c'}\right)\bar{\mathrm{u}}_c\ \ ,\ \ \bar{\mathrm{u}}_c(0)=\bar{\mathrm{u}}_c^0 \tag{10}$$

The first term on the right hand side indicates the increase of $\bar{\mathrm{u}}_c$ due to the agents coming from adjacent locations to c, and the second term indicates the decrease of $\bar{\mathrm{u}}_c$ due to c agents leaving for the adjacent locations.

By the Cauchy-Lipschitz theorem, for any initial condition $\bar{\mathrm{u}}^0 = \langle\bar{\mathrm{u}}_c^0\rangle_{c\in L}$, Equation 10 admits a unique solution [11]. Let $\bar{\mathrm{u}}_c(t \mid \bar{\mathrm{u}}^0)$ denotes the deterministic value of $\bar{\mathrm{u}}_c$ at time t given the initial condition $\bar{\mathrm{u}}^0$. The stationary location occupancy measure can be derived using the fixed point method:

$$\forall c{\in}L,\frac{\partial\ \bar{\mathrm{u}}_c(t)}{\partial t}=0\implies\forall c\in L,\tilde{u}_c\left(\sum_{c'\neq c}\rho_{c',c}u_{c'}\right)=\left(\sum_{c'\neq c}\rho_{c,c'}\right)\tilde{u}_c\ ,\ \sum_{c\in C}\tilde{u}_c=1.$$

Evolution of Age Distributions. Consider $\boldsymbol{M}^N$, the state vector stated above. Assume that there are N agents in the network. The system's occupancy measure is defined as $\bar{\mathbf{M}}^N(t) = \frac{\boldsymbol{M}^N(t)}{N} = \langle\, \bar{M}_1(\cdot,t), \bar{M}_2(\cdot,t), \ldots, \bar{M}_C(\cdot,t)\,\rangle$, where $\forall c \in L, \bar{M}_c^N(z,t)$ denotes the *density* of agents in c with age z at time t. We also define $F_c^N(z,t)$, the cumulative distribution function over $\bar{M}_c^N(t)$:

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

$\forall c \in L, \forall z, t \in \mathbb{R}^+$, $F_c^N(z,t)$ shows the proportion of N in c with age *less than or equal* to z. We assume that when $N \to \infty$, the initial occupancy measures $\bar{\mathbf{M}}^N(0)$ converge to a unique limit $\bar{\mathbf{m}}^0$: $\lim_{N\to\infty} \bar{\mathbf{M}}^N(0) = \bar{\mathbf{m}}^0$. This implies that $\forall c \in L\ ,\ \lim_{N\to\infty} \bar{\mathrm{M}}_c^N(0) = \bar{\mathrm{m}}_c^0$.

The rate functions related to the data propagation satisfy the density dependence condition. Therefore, for any $t > 0$ and for all $c \in L$, when $N \to \infty$, $\bar{\mathrm{M}}_c^N(t)$ converges to $\bar{\mathrm{m}}_c(t)$, where $\bar{\mathrm{m}}_c(t)$ is the solution of the following partial differential equation [11]. Here, $\bar{\mathrm{u}}_c(t)$ is derived by solving Equation (10) for t.

$$\bar{\mathrm{m}}_c(0,t) = \mu_c \times \bar{\mathrm{u}}_c(t) \tag{11}$$

$$\frac{\partial \bar{\mathrm{m}}_c(z,t)}{\partial t} = -\frac{\partial \bar{\mathrm{m}}_c(z,t)}{\partial z} - \mu_c \bar{\mathrm{m}}_c(z,t) + \sum_{c' \neq c} \rho_{c',c} \bar{\mathrm{m}}_{c'}(z,t) - \left(\sum_{c' \neq c} \rho_{c,c'}\right) \bar{\mathrm{m}}_c(z,t) \tag{12}$$

$$+2\eta_c \left[(+1) \times (u_c(t) - F_c(z,t)) \cdot \bar{\mathrm{m}}_c(z,t) + (-1) \times \bar{\mathrm{m}}_c(z,t) \cdot F_c(z,t)\right]$$
$$+ \sum_{c' \neq c} 2\beta_{c,c'} \left[(+1) \times (u_c(t) - F_c(z,t)) \cdot \bar{\mathrm{m}}_{c'}(z,t) + (-1) \times \bar{\mathrm{m}}_c(z,t) \cdot F_{c'(z,t)}\right]$$

We propose an intuitive explanation for forming Equation 12 by considering how much each $\bar{\mathrm{m}}_c(z,t)_{c\in C}$ changes in an small time interval ∂t (the left hand side). Consider $c \in L$. During ∂t, agents with age z (accounted for by $\bar{\mathrm{m}}_c(z,t)$) grow older and need to be removed from $\bar{\mathrm{m}}_c(z,t)$. Additionally, agents with age $z - \triangle z$ become older and the density $\bar{\mathrm{m}}_c(z - \triangle z, t)$ need to be added to $\bar{\mathrm{m}}_c(z,t)$. Hence, the rate of change of $m_c(z,t)$ caused only by *aging* is (first term on the right hand side of Eq. 12):

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

The second term reflects the communication of agents, accounted by $\bar{\mathrm{m}}_c(z,t)$, with one of the base stations. If, there are $\bar{\mathrm{m}}_c(z,t)$ agents in c, given that the rate of communication with base stations in c is μ_c, then in ∂t, $\mu_c \times \bar{\mathrm{m}}_c(z,t) \times \partial t$ communications take place and the agents involved leave $\bar{\mathrm{m}}_c(z,t)$. Therefore, the rate of the change is $\mu_c \times \bar{\mathrm{m}}_c(z,t)$.

The third expression shows the increase of $\bar{\mathrm{m}}_c(z,t)$ as a result of agents with age z moving from other locations c' into c. The rate of the increase due to the flow from any $c' \neq c$ is $\rho_{c,c'} \bar{\mathrm{m}}_{c'}(z,t)$. Conversely, the fourth term reflects the movement of agents contained in $\bar{\mathrm{m}}_c(z,t)$ out of c into the adjacent locations. The decrease in $\bar{\mathrm{m}}_c(z,t)$ due to this flow happens at rate $\sum_{c'\neq c}\rho_{c,c'}$.

The fifth term has two parts. The first shows the rate of the flow into $\bar{\mathrm{m}}_c(z,t)$ due to agents with age z in c communicating with agents of higher age in c. The total density of agents in c at time t is $\bar{\mathrm{u}}_c(t)$and of those with age less than z is $F_c(z,t)$. Therefore, $(u_c(t)-F_c(z,t))$ is the density of agents older than z. In the normalised system, by Equation 9, the rate of communication between the fraction with age z and those with higher ages is: $2\eta_c(u_c(t)-F_c(z,t))\bar{\mathrm{m}}_c(z,t)$. The second part, $-2\eta_c(\bar{\mathrm{m}}_c(z,t))F_c(z,t)$, reflects the drift out of $\bar{\mathrm{m}}_c(z,t)$ as a result of agents with age z in c communicating with agents of lower age in c.

The sixth term is similar to fifth, with the difference that it shows the change of $\bar{\mathrm{m}}_c(z,t)$ due to the agents from c communicating with agents from $c' \neq c$.

We simplify Equation 12 by integrating over z to obtain:

$$\forall c\in L\colon \frac{\partial\ F_c(z,t)}{\partial t} = -\frac{\partial\ F_c(z,t)}{\partial z} + \left(\sum_{c'\neq c}\rho_{c',c}\ F_{c'}(z,t)\right) - \left(\sum_{c'\neq c}\rho_{c,c'}\right) F_c(z,t) \qquad (13)$$

$$+\big(u_c(t|d)-F_c(z,t)\big)\big(2\eta_c F_c(z,t)+\mu_c\big) + \big(u_c(t|d)-F_c(z,t)\big)\sum_{c'\neq c} 2\,\beta_{c,c'} F_{c'}(z,t)$$

$$\forall c\in L, \forall t\geq 0 : F_c(0,t)=0 \quad , \quad \forall c\in L, \forall z\geq 0 : F_c(z,0)=F_c(z)$$

In this modelling, the set of ODEs (10) are constructed and solved independently, as the agents' mobility is not assumed to be dependent on the data propagation.

4.3 Solution of the Equations

Here we consider how Equation 13 is solved, for the case where there is only *one location* in the system and at $t=0$, every agent has age zero.

The solution is obtained by introducing a change of variables. Let the space $\mathcal{A}=\{(x,y)\in\mathbb{R}\times\mathbb{R}|\, x\geq 0, x+y\geq 0\}$ and $G(x,y): A\to[0,1]$, $G(x,y)=F(x,x+y)$. In order to find $F(z,t)$ it is enough to derive $G(z,t-z)$. For function G we have:

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

Rearranging the terms in Equation (13), we obtain:

$$\frac{\partial G(x,y)}{\partial x} = (1-G(x,y))(2\eta\, G(x,y)+\mu) \qquad G(0,y)=0 \qquad (14)$$

The assumption that at time $t=0$, no gossip exists, implies that $\forall t\ z<t$ and $y=t-z>0$. For anu $y\in\mathbb{R}^+$, let us define $g_y : x\mapsto G(x,y)$. Therefore:

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

By Cauchy-Lipschitz Theorem, this equation has a solution. The value obtained for $g_y(x)$ leads to the corresponding $F(z,t)$.

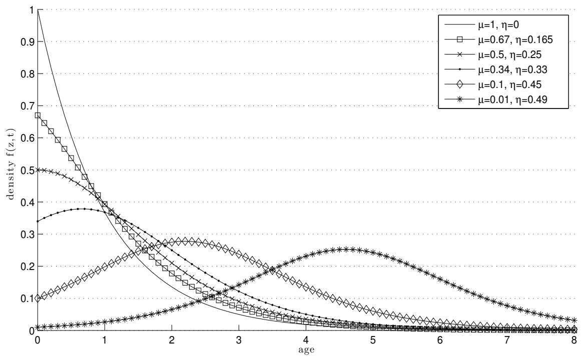

Fig. 6: the density at age z for different values of η and μ when $z \leq t$.

Single Location - Analytical Solution. In this case, Equation 14 can be analytically solved to obtain the following solution:

$$F(z,t) = \begin{cases} 1 - \dfrac{2\eta + \mu}{2\eta + \mu e^{(\mu+2\eta)z}} & if\ z \leq t \\ \\ 1 - \dfrac{2\eta + \mu}{2\eta + \frac{2\eta F(z-t,0)+\mu}{1-F(z-t,0)} e^{(\mu+2\eta)t}} & if\ z > t \end{cases} \tag{15}$$

We illustrated the reasoning behind the first case of the solution (when $z \leq t$). The second case ($z > t$), concerns the situation where in the initial configuration some agents have age greater than zero. Therefore, at any time t, it is possible to have agents with ages higher than t. The proportion of the agents who at time t have age $z > t$ depends on the proportion whose age was at least $(z - t)$ in the system's initial configuration. We skip the solution explanation for this case.

The solution allows us to study important aspects of the peer to peer network. In terms of performance, the network is well designed if with a high probability, the majority of agents remain within relatively low ranges of age. One way to satisfy this performance requirement is to deploy a relatively large number of base stations in each location; the agents frequently communicate with the base stations and receive fresh copies of the data. We introduce the term *infrastructure dominant* here. A location where the associated age distribution is mainly formed by the agent-to-base-station communication is said to be infrastructure dominant. In such a location, the agent-to-agent communication has less impact.

A location that does not enjoy strong infrastructure may still exhibit a satisfactory age distribution. In this case, the frequent and improved agent-to-agent communication is the main contributing factor in information dissemination. A location where the opportunistic contact determines the shape of the age distribution is referred to as *opportunistic 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 conclude [11] that when $\mu \geq 2\eta$, $m(z,t)$ decreases as the age increases. The maximum density is at age $z = 0$ with $m(0,t) = \mu$. Here, the opportunistic contacts happen at a lower rate than with the base stations. Hence, the latter type of communication determines the shape of the distribution. The extreme case is when $\eta = 0$; the opportunistic contact does not occur at all. In this case, improving the age distribution entails improving the rate of communications with base stations by increasing the number of base stations.

We also conclude that when $\mu < 2\eta$, the opportunistic contact rate becomes large enough to influence the age distribution. Consequently, there emerges a large mass around a *typical* age, maintained by the contacts 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 centred at some age $z > 0$. In order to improve the age distribution in such a network without changing μ, one needs to improve η.

Multiple Locations. We explain the steps in the solution phase when the network contains multiple locations. Let us assume that the system has reached its equilibrium; $\forall c \in L$, $\frac{\partial F_c(z,t)}{\partial t} = 0$, $u_c(t) \to \tilde{u}_c$. Using Equation (13), we obtain:

$$\forall c \in L,\ \frac{d\ F_c(z)}{dz} = +\tilde{u}_c\mu_c + \left(\tilde{u}_c 2\eta_c - \mu_c - \sum_{c' \neq c} \rho_{c,c'}\right) F_c(z) \tag{16}$$

$$+ \sum_{c' \neq c} (\rho_{c',c} + \tilde{u}_c 2\beta_{c,c'}) F_{c'(z)} - \sum_{c' \neq c} 2\beta_{c,c'} F_c(z).F_{c'(z)} - 2\eta_c (F_c(z))^2$$

with the initial condition $\forall c \in L,\ F_c(0) = 0$. In contrast with the previous case, this system of ODEs is multi-dimensional and non-linear, and has no simple analytical solution. Nevertheless, when $z \to 0$ or z is very large, it can be approximately solved. If $z \to 0$, then $F_c(z) \to 0$ and the factors $F_c(z)F_{c'}(z)$ and $(F_c(z))^2$ become negligible compared to the rest of the expression and can be ignored to find the following system shown in the matrix form:

$$F' = FA + B \tag{17}$$

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

For $c \in L$ and $z \to 0$, $m_c(z) \approx \mu_c \tilde{u}_c$. The derivative of $m_c(z)$ is:

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

If $\forall\ c, c' \in L : \beta_{c,c'=0}$, then:

$$\frac{d\ \bar{\mathrm{m}}_c(z)}{dz} = \mu_c \tilde{u}_c (\tilde{u}_c 2\eta_c - \mu_c) + \sum_{c' \neq c} (\mu_{c'} - \mu_c) \tilde{u}_{c'} \rho_{c',c} \tag{18}$$

Equation (18) is used to determine for each c, whether its is an infrastructure dominant or opportunistic contact dominant. If $\forall c, \mu_c = \mu$ (μ_c is the same in all locations), c 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 c 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 any location with no base stations, 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 this case, the nature of each location can be decided only after plugging the parameters into Equation (18) 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 simplify the equations [11].

4.4 Model Validation

We reviewed how the model was developed and analysed [11]. Now we focus on *model validation.* This task has three steps. First, by using the data on the executions of the real system (eg. time series) the model's parameters are found. Then, a version of the model with concrete values for the parameters is constructed. Second, using a classical approach such as the stochastic simulation, the model is analysed and the observations are compared (qualitatively/quantitatively) against the real executions to check whether the model effectively captures the age distributions. Finally, the mean-field solution is obtained to check whether this particular method is suitable for the analysis of the model.

Validation Platform. CabSpotting [10] is a project where the San Francisco taxi company traces the location of its yellow cabs as they operate in the Bay Area (SFBA). Using GPS, each cab reports its location every minute and the data is stored in a database. By using the cabs' movement traces and introducing some realistic networking assumptions, one can construct a realistic opportunistic peer-to-peer network, similar to the model considered in Section 4.1, where the cabs and base stations are responsible for propagating data in the network. The realistic scenario, built in this manner, is used in the model validation.

Assume that SFBA is divided into 16 locations. There are a number of base stations which frequently transmit fresh copies of a piece of data. Each base station has a specific transmission range. The network consists also of a relatively larger number of taxi cabs. Each cab is equipped with a radio device to communicate with base stations or other cabs. Each cab scans its surrounding once per minute and when another entity is detected (another cab or one of the base stations), it tries to initiate a data exchange. The radio devices are assumed to have the range of 200m. A *meeting* or successful data exchange happens if the communicating entities remain in 200-meter proximity for at least 10 seconds (10 sec guarantees a data exchange). The goal of the meetings is to propagate updated copies of the data throughout the network. The age of a cab is equal to the time elapsed since the data it holds was sent by one of the base stations.

The CabSpotting database stores the cabs' movement traces. By using these traces and making the networking assumptions stated above, we can generate the *contact traces*. The latter not only captures the occurrence of the meetings, but also how the age of the cabs change as the result of such meetings. Therefore, the contact traces record how the cabs' ages change and can be used to observe how the age distributions evolve in different locations. In [11], 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. They were then used for validation steps.

Extracting Model Parameters. The following quantities were measured using the contact traces generated. $N(t)$: total number of cabs in time slot t (time unit = one minute); $N_c(t)_{c\in\{1,2,3,\ldots 16\}}$: number of cabs in location c during time slot t; $N_{c,ub}(t)$: number of contacts between a mobile agent and a base station in c during time slot t; $N_{c,uu}(t)$: number of contacts between any two mobile agents in c during t; $N_{c,c',uu}(t)_{c\neq c'}$: number of contacts between an agent from c and another from c' during t.

Given the contact traces, one can calculate $\bar{\mu}_c(t) = \frac{N_{c,ub}(t)}{N_c(t)}$ as the rate at which an agent in c communicates with one of the base stations in that location during t. If, at t there are $N_c(t)$ agents in c, then on average $\bar{\mu}_c(t) \times N_c(t)$ meetings are expected in the following time unit. The average μ_c for an hour is calculated by averaging $\bar{\mu}_c(t)_{t\in[0,59]}$: $\mu_c = \frac{1}{60}\sum_{t=t_0}^{t_0+59} \bar{\mu}_c(t)$. This parameter is used in the model. Let us now focus on how other parameters are calculated.

In the model, for $c \in L$ the rate at which an agent in c meets another agent in c is $\frac{2\eta_c}{N-1}$. Consequently, the rate at which meetings occur in c is:

$$\binom{N_c}{2} \times \frac{2\eta_c}{(N-1)} = \frac{(N_c)\times(N_c-1)}{N-1} \times \eta_c. \tag{19}$$

During the time unit t, the traces capture $N_{c,uu}(t)$ meetings which can be expressed using Equation 19. We assume that that at t, $\bar{\eta}_c(t)$ affects the rate of the meetings. Therefore, in time unit t we expect to observe $\frac{N_c(t)\times(N_c(t)-1)}{(N(t)-1)}\bar{\eta}_c(t)$ meetings. Thus:

$$N_{c,uu}(t) = \frac{(N_c(t))(N_c(t)-1)}{N(t)-1}\bar{\eta}_c(t) \Rightarrow \bar{\eta}_c(t) = \frac{N_{c,uu}(t)}{\frac{N_c(t)}{N(t)-1}(N_c(t)-1)} \approx \frac{N_{c,uu}(t)}{u_c(t)(N_c(t)-1)}.$$

The model's η_c is obtained by averaging $\bar{\eta}_c(t)$ for one hour; $\eta_c = \frac{1}{60}\sum_{t=t_0}^{t_0+59} \bar{\eta}_c(t)$.

In the model, the rate at which an agent in c meets an agent in c' is $\frac{2\times\beta_{c,c'}}{N-1}$. Therefore, in one time unit, on average $\frac{2\beta_{c,c'}}{N-1}N_cN'_c$ meetings occur between agents in c and c'. For each time unit t, the traces show $N_{c,c',uu}(t)$ meetings having occurred. Therefore:

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

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

For each c and c', $\beta_{c,c'}$ can is obtained by averaging $\bar{\beta}_{c,c'}(t)$ over an hour.

Finally, in the model, the rate at which agents move from location c to c' is defined to be $\rho_{c,c'} \times N_c$. In the traces, one observes $N_{c,c',trans}(t)$ movements. Therefore: $\bar{\rho}_{c,c'}(t) N_c(t) = N_{c,c',trans}(t) \Rightarrow \bar{\rho}_{c,c'}(t) = \frac{N_{c,c',trans}(t)}{N_c(t)}$. The same averaging is applied to $\bar{\rho}_{c,c'}(t)$ to find $\rho_{c,c'}$.

The parameters obtained from the contact traces were used to build a fully parametrized model. The model was then simulated and the stochastic behaviour obtained was compared against the traces. The authors show that the model is sufficiently detailed to capture the stochastic behaviour of the real system.

The last step of the validation is checking whether the mean-field method is an appropriate method for the analysis of this model. The authors show that for the locations which usually have reasonably large populations of agents (having at least tens of taxi cabs), there exists a close correspondence between the age distributions obtained from the mean-field analysis and the distributions derived from the contact traces. For the locations at the edges of the network, where the population of the cabs were too small, the mean-field solution has more error. Due to space limitation we skip reviewing the last sections of the validation process and the interested reader is referred to [11].

5 Model Checking Mean-Field models

In this section we discuss model-checking approach for mean-field models. The kind of analysis we can perform through model checking is rather different from the performance studies we illustrated in previous sections. 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.

There are two possible ways of describing the properties of a large population: via studying a random individual within the whole population and via considering the whole population.

The first approach is known as a *fluid model checking* [8] and it employs a bounded fragment of the *Continuous Stochastic Logic* (CSL) for describing properties of interest. Later in this section we recall the logic CSL, and explain how these properties can be checked for an individual object.

While fluid model checking is applicable to the local model only, the second approach allows us to derive the properties of the overall mean-field model. This is done using *Mean-Field Continuous Stochastic Logic* (MF-CSL) [25], which lifts the properties of the local model to the level of the overall model via *expectation operators*. MF-CSL logics relays on the local model properties when constructing the properties of the overall model, and the timed properties can be described only on the local level (for an individual object).

Note that yet another approach to model-checking mean-field models is possible, that only makes use of the deterministic limit (occupancy vector) to reason about the timed properties on the level of the overall model.

In the following we first return our attention to the single agent and its properties in Sections 5.1-5.5. Then the model-checking procedure for the whole population is addressed in Section 5.6.

5.1 Single Agent Model

An interesting consequence of the mean-field approximation theorem is the so-called *decoupling of joint probability* (for details, please refer to [3,30]), which allows us to obtain the model of the single object within the overall model, by using *fast simulation* [13,15]. The central idea of this process is to abstract the system to its fluid approximation (to obtain mean-field model of the system) and to study the evolution of a single agent as executed in parallel with the approximation of the rest of the system. 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 the dynamics of a single agent depend on the other agents only through the overall average system state. This allows us to reason about the local model within the overall model as of a time-*in*homogeneous continuous time Markov chain (ICTMC).

Due to the time-inhomogeneity of the local model, the existing model checking algorithms for CTMCs can not be reused. Therefore, in [8] the authors develop novel CSL model checking algorithms for ICTMC models. We denote the single object model coupled with the deterministic limit (the local ICTMC) as $Z(t)$ for ease of notation. The labelling of the states of ICTMC is done on the same way as for a time-homogeneous CTMC.

5.2 Continuous Stochastic Logic

As a single agent model is described by an ICTMC, a standard CSL logic can be used to express the properties of such model. In the following we recall the definition of *bounded* CSL as in [2]:

Definition 3. ***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 CSL formulas we first recall the notion of a path as it was defined for the CTMCs in [2]; this notion is reused for ICTMCs. An *infinite path* σ is a sequence $s_0 \overset{t_0}{\rightarrow} s_1 \overset{t_1}{\rightarrow} s_2 \overset{t_2}{\rightarrow} ...$, for $i \in \mathbb{N}$; $s_i \in S$ 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 $s_0 \overset{t_0}{\rightarrow} s_1 \overset{t_1}{\rightarrow} ...s_{l-1} \overset{t_{l-1}}{\rightarrow} s_l$ 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 given 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 $\delta(\sigma; i)$. Moreover, with i the smallest index, and 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 $t > \sum_{j=0}^{l-1} t_j$. $Path^{Z(t)}(s_i, t_0)$ is the set of all finite and infinite paths of the ICTMC that start in state s_i and $Path^{Z(t)}(t_0)$ includes all (finite and infinite) paths of the ICTMC. A probability measure $Pr(t_0)$ 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 4. ***Semantics of CSL.*** *Satisfaction of state and path CSL formulas for ICTMCs is given as follows:*

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

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

Only *bounded* time intervals are used in path formulas. This is motivated by the nature of convergence theorem, which is valid only for finite-time horizons. The relaxation of this restriction is possible, but we will not discuss it this tutorial, see [8], and [25] for details.

The CSL operators can be nested according to Definition 3. 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].

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 (see [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]. The way to calculate this probability will be presented below. 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 $s_1 \in \mathbb{S}$ at the current time and reaching a goal state $s_2 \in \mathbb{G}$ 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. 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. 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 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 $t_2 - t_1$.

$$\mathrm{Prob}^{\mathcal{A}}(s, \Phi_1 U^{[t_1,t_2]}\Phi_2) = \sum_{s_1 \models \Phi_1} \sum_{s_2 \models \Phi_2} \pi^{\mathcal{A}[\mathbb{U}]}_{s,s_1}(t_1) \cdot \pi^{\mathcal{A}[\mathbb{U}\cup\mathbb{G}]}_{s_1,s_2}(t_2 - t_1). \tag{20}$$

Hence, CSL until formulas can be solved as a combination of two reachability problems, as shown in Equation (20), 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 agent, 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.

5.3 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 an individual agent. 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. Another important difference between checking CSL formulas for CTMC and ICTMC is in the fact that the set of goal states (states, which satisfy Φ) can change with time. In the following we address these differences and explain how a bounded CSL Next formula 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, i.e., the probability to jump from the state s to the state, satisfying Φ, or goal state, within time interval $[t_1, t_2]$. This probability can be found as follows:

$$\mathrm{Prob}^{Z(t)}(s, \mathcal{X}^{[t_1,t_2]}\Phi_2, t_0) = \int_{t_0+t_1}^{t_0+t_2} q_{s,\mathbb{G}}(t) \cdot e^{-\Lambda(s,t_0,t)} dt, \qquad (21)$$

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 [8].

The next state probability can now be computed numerically in two ways: using Equation (21) or by transformation the above formula to the differential equations and solving them. The differential equations are more convenient and simplify the calculations, they can be obtained as in [8], and are as follows:

$$\begin{cases} \dot{P}(t) = q_{s,\mathbb{G}}(t) \cdot e{-}L(t), \\ \dot{L}(t) = -q_{s,s}(t), \end{cases} \qquad (22)$$

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 $\mathrm{Prob}^{Z(t)}(s, \mathcal{X}^{[t_1,t_2]}\Phi_2, t_0, t)$ is needed to be accessed. To find this dependency one has to either calculate integral (21) for all possible t_0, or use the differential equations (22) to define another system of the differential equations with t_0 as a independent variable. The obtained new system of differential equations is as follows:

$$\begin{cases} \dot{\overline{P}}_s(t) = q_{s,\mathbb{G}}(t + t_2) \cdot e{-}L_2(t) - q_{s,\mathbb{G}}(t + t_1) \cdot e{-}L_1(t) - q_{s,s}(t)\overline{P}_s(t), \\ \dot{L}_1(t) = -q_{s,s}(t) + q_{s,s}(t + t_1), \\ \dot{L}_2(t) = -q_{s,s}(t) + q_{s,s}(t + t_2), \end{cases} \qquad (23)$$

where $L_1(t) = \Lambda(t, t + t_1)$ and $L_2(t) = \Lambda(t, t + t_2)$. Initial conditions at time t_0 are computed by solving Equation (22).

And finally, 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 (23) is solved for each time interval $[T_i, T_{i+1}]$.

For checking next formula one has to compare next state probability with the given threshold $p \in [0, 1]$, hence, equation $\mathrm{Prob}^{Z(t)}(s, \mathcal{X}^{[t_1,t_2]}\Phi_2, t_0, t) = p$ 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 *piecewise real analytical functions*, as described and proved in [8].

5.4 Until formulas. Reachability Probability

The core idea of CSL model-checking of until formulas as explained in Section 5.2 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 $\pi^{Z(t)}_{s,s_1}(t', T)$ 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 ICTMCs $Z(t)[\mathbb{U}]$ and $Z(t)[\mathbb{U} \wedge \mathbb{G}]$, respectively:

$$\mathrm{Prob}^{Z(t)}(s, \Phi_1 U^{[t_1,t_2]}\Phi_2, t') = \sum_{s_1, t' \models \Phi_1} \sum_{s_2, t_1 \models \Phi_2} \pi^{Z(t)[\mathbb{U}]}_{s,s_1}(t', t_1 - t') \cdot \pi^{Z(t)[\mathbb{U}\wedge\mathbb{G}]}_{s1,s2}(t_1, t_2 - t_1). \tag{24}$$

Equation (24) is valid for $t_1 > t', t_2 > t'$. If $t_1 = t'$ the first reachability problem can be omitted.

In the following we explain here how an arbitrary reachability probability $\Pi'(t', t' + T)$ can be calculated. This method is applied to both $\pi^{Z(t)[\mathbb{U}]}_{s,s_1}(t', t_1 - t')$ and $\pi^{Z(t)[\mathbb{U}\wedge\mathbb{G}]}_{s1,s2}(t_1, t_2 - t_1)$; and the results are combined as in (24). 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 identity matrix as initial condition:

$$\frac{d\Pi'(t', t' + T)}{d(T)} = \Pi'(t', t' + T) \cdot Q'(t' + T), \tag{25}$$

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

In order to check a nested 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:

$$\frac{d\Pi'(t, t + T)}{dt} = -Q'(t)\Pi'(t, t + T) + \Pi'(t, t + T)Q'(t + T). \tag{26}$$

The time-dependent probability matrix $\varPi'(t, t+T)$ can be obtained by solving Equation (26) with initial condition $\varPi'(t', t'+T)$. Using Kolmogorov equations for solving reachability problems on the local models $Z(t)$ is efficient due to the fact that the state space is usually quite small (see [8]).

The goal and unsafe sets in ICTMC can vary with time (e.g., in nested formulas), which has to be taken into account while calculating reachability probability. This is done by solving Equation (26) 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 $i = 0, ..., k$.

The procedure has to be slightly adjusted to ensure that only safe states are visited before a goal state is reached. We need to modify the ICTMC $Z(t)$ for each time interval $(T_i; T_{i+1})$ as follows:

1. introduce a new goal state s^*, which remains the same for all time intervals;
2. all unsafe and goal states are made absorbing;
3. all transitions leading to goal states are readdressed to the new state s^*.

Given this modified ICTMC $\overline{Z(t)}$, the transient probability matrix $\overline{\varPi'}(T_i, T_{i+1})$ is found for each time interval using the forward Kolmogorov equation, according to Equation (25).

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 any-more. 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 $(|S|+1) \times (|S|+1)$ constructed in the following way: for each state $s \in S$ which is safe before and after T_i it follows $\zeta(T_i)_{s,s} = 1$. For each state $s \in S$ which was safe before T_i and becomes 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 $\varUpsilon(t', t'+T)$:

$$\begin{aligned}\varUpsilon(t', t'+T) =& \overline{\varPi'}(t', T_1) \cdot \zeta(T_1) \cdot \overline{\varPi'}(T_1, T_2) \cdot \\ & \zeta(T_2) \ldots \zeta(T_k) \cdot \overline{\varPi'}(T_k, t'+T).\end{aligned} \tag{27}$$

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

$$\begin{aligned}\pi^{[\mathbb{U} \vee \mathbb{G}]}_{s,s^*}(t', t'+T) =& \varUpsilon_{s,s^*}(t', t'+T) + \\ & \mathbf{1}\{s \in Sat(\mathbb{G}, t')\}.\end{aligned} \tag{28}$$

Similarly to the dependency on time of the reachability probability while the goal and unsafe sets are fixed (see Equation (26)), 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 (see [8] for more details).

The method for checking state and path CSL formulas for ICTMC was presented above in this section. The convergence of the results and decidability of the algorithms are addressed in [8]. This method is applicable for the continuous time models, as the main interest of this tutorial lies in a continuous time mean-filed models. For the similar results on the *on-the-fly* fast model-checking of the PCTL properties of the individual objects in a discrete time mean-field model we refer to [29]. As a next step we provide the example, where this method is applied to a single agent of mean-field model.

5.5 Examples

In this section couple of examples of checking CSL formulas are described. We reuse the virus spread model, described in the Examples 1 and 2 (see Figure 1). As descibed in Section 2, the system of the limit ODEs (6) for the population behaviour is as follows:

$$\begin{cases} \dot{x}_1(t) = -k_1 x_3(t) + k_2 x_2(t) + k_5 x_3(t), \\ \dot{x}_2(t) = (k_1 + k_4) x_3(t) - (k_2 + k_3) x_2(t), \\ \dot{x}_3(t) = k_3 x_2(t) - (k_4 + k_5) x_3(t). \end{cases} \tag{29}$$

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 $x_1(t)$ and $x_3(t)$ are the solution of the ODEs (29) with $\overline{x}(0)$ as initial condition. Therefore the transition rate matrix $\mathbf{Q}(\overline{x}(t))$ is as follows:

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

To find $Prob^{Z(t)}(s, \text{not infected } U^{[0,1]} \text{ infected}, t')$ only one reachability problem $\pi_{s,s_1}^{Z(t)[\neg\text{not infected}\vee\text{infected}]}(0,1) = \pi_{s,s_1}^{Z(t)[\text{infected}]}(0,1)$ has to be solved according to the algorithm described earlier in Section 5.4. The local model $Z(t)$ 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:

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

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.

The probability of the until formula

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

to hold for each starting state is as follows:
$Prob^{Z(t)}(s_1, \phi, t') = \pi_{s_1,s_2}^{Z(t)[\text{infected}]}(0,1) + \pi_{s_1,s_3}^{Z(t)[\text{infected}]}(0,1) = 0.09;$
$Prob^{Z(t)}(s_2, \phi, t')) = 0;$
$Prob^{Z(t)}(s_3, \phi, t')) = 0.$

The found above probabilities are compared with 0.3, and 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 $Prob^{Z(t)}(s, \text{not infected } U^{[0,1]}\text{infected}, t', t)$ is done as described earlier in this section:

1. the model $Z(t)$ is modified so the infected states are made absorbing;
2. the transient probability $\mathit{\Pi}'(0,1)$ is calculated as described in the example above;
3. 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 (26)).
4. These ODEs are solved using $\mathit{\Pi}'(0,1)$ as initial condition. The solution of the ODEs defines the required reachability probabilities.

The time-dependent probability $Prob^{Z(t)}(s_1, \text{not infected } U^{[0,1]} \text{ infected}, t', t)$ is depicted in Figure 7. 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 $Prob^{Z(t)}(s_1, \text{not infected } U^{[0,1]} \text{ infected}, t', t) = 0.3$ 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 a 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:

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

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 $Prob^{Z(t)}(s, tt\ U^{[0,0.5]} \text{ infected}, t', t)$ is calculated for all states $s \in S^o$. In Figure 7 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*. Similarly to the previous example, the time dependent satisfaction set is found and equals to $Sat(\Phi_1, t', t) = \{s_2, s_3\}$ for all $t \in [0, 10.443]$ and $Sat(\Phi_1, t', t) = \{s_1, s_2, s_3\}$ for all $t \in (10.443, 15]$.

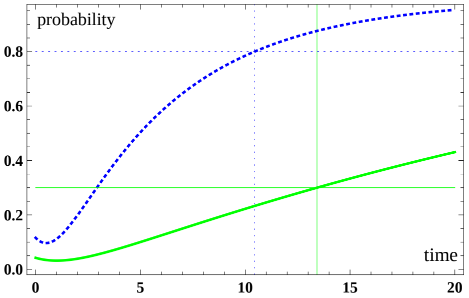

Fig. 7: The green solid line shows $Prob^{Z(t)}(s_1, \text{not infected } U^{[0,1]} \text{ infected}, t', t)$. The time-dependent probability $Prob^{Z(t)}(s_1, tt\ \ U^{[0,0.5]} \text{ infected}, t', t)$ is presented by the blue dotted line.

The next task is calculating the probability

$$Prob^{Z(t)}(s, \text{infected } U^{[0,15]}\Phi_1, t', t).$$

The reachability probability for the time-varying satisfaction set of Φ_1 is calculated following the algorithm described 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 $s_1 \neq s^*$,$s_2 \neq s^*$. The transient probabilities on time intervals $[0, 10.443)$ and $(10.443, 15]$ are calculated using the 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 (27) 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 (28) is used in order to calculate the reachability probability for each state $s \in S^o$: $\pi_{s_1,s^*}^{Z(t)[\neg\text{infected}\vee\Phi_1]}(0,15) = 0.47$; $\pi_{s_2,s^*}^{Z(t)[\neg\text{infected}\vee\Phi_1]}(0,15) = 1$; $\pi_{s_3,s^*}^{Z(t)[\neg\text{infected}\vee\Phi_1]}(0,15) = 1$. The probability $Prob^{Z(t)}(s, \text{infected } U^{[0,15]}\Phi_1, t')$ is calculated according to Equation (24), 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 have illustrated how the properties of a single agent in a large communication network (system of interacting objects) can be checked. Next to the fluid model checking reader might be interested in the techniques for calculation *fluid passage time*, as discussed in [18]. In the following model-checking the overall mean-field model is discussed.

5.6 On Model-Checking Overall Mean-Field Models. MF-CSL.

The properties of interest of the overall mean-field model differ from the properties which can be described by CSL. Therefore, in order to reason at the level of the overall model in terms of fractions of objects an extra layer "on top of CSL" that defines the logic MF-CSL was introduced in [25]. The latter is able to describe the behaviour of the overall system in terms of the behaviour of random local objects.

Definition 5. ***Syntax of MF-CSL.*** *Let $p \in [0,1]$ be a real number, and $\bowtie \in \{\leq, <, >, \geq\}$ a comparison operator. MF-CSL formulas Ψ are defined as follows:*

$$\Psi ::= tt \mid \neg\Psi \mid \Psi_1 \wedge \Psi_2 \mid \mathbb{E}_{\bowtie p}(\Phi) \mid \mathbb{ES}_{\bowtie p}(\Phi) \mid \mathbb{EP}_{\bowtie p}(\phi),$$

where Φ is a ***CSL state formula*** *and ϕ is a* ***CSL path formula****.*

□

In this definition three expectation operators were introduced: $\mathbb{E}_{\bowtie p}(\Phi)$, $\mathbb{ES}_{\bowtie p}(\Phi)$ and $\mathbb{EP}_{\bowtie p}(\phi)$, with the following interpretation:

- $\mathbb{E}_{\bowtie p}(\Phi)$ denotes whether the fraction of objects that are in a (local) state satisfying a general CSL state formula Φ fulfills $\bowtie p$;
- $\mathbb{ES}_{\bowtie p}(\Phi)$ denotes whether the fraction of objects that satisfy Φ in steady state, fulfills $\bowtie p$;
- $\mathbb{EP}_{\bowtie p}(\phi)$ denotes whether the probability of a random object to satisfy path-formula ϕ fulfills $\bowtie p$.

The formal definition of the MF-CSL semantics is as follows:

Definition 6. ***Semantics of MF-CSL.*** *The satisfaction relation $\models$ for MF-CSL formulas and states $\overline{x} = (x_1, x_2, \ldots, x_K)$ at time t_0 of the overall mean-field model is defined by:*

$$\begin{array}{ll}
\overline{x} \models tt & \forall \ \overline{x} \ \in \ X, \\
\overline{x} \models \neg \Psi & \text{iff } \overline{x} \not\models \Psi, \\
\overline{x} \models \Psi_1 \wedge \Psi_2 & \text{iff } \overline{x} \models \Psi_1 \wedge \overline{x} \models \Psi_2, \\
\overline{x} \models \mathbb{E}_{\bowtie p}(\Phi) & \text{iff } \left(\sum_{j=1}^{K} x_j \cdot Ind_{(s_j, t_0 \models \Phi)} \right) \bowtie p, \\
\overline{x} \models \mathbb{ES}_{\bowtie p}(\Phi) & \text{iff } \left(\sum_{j=1}^{K} x_j \cdot \pi^{Z(t)}(s_j, Sat(\Phi, t_0)) \right) \bowtie p, \\
\overline{x} \models \mathbb{EP}_{\bowtie p}(\phi) & \text{iff } \left(\sum_{j=1}^{K} x_j \cdot Prob^{Z(t)}(s_j, \phi, t_0) \right) \bowtie p,
\end{array}$$

where $Sat(\Phi, t_0)$ is a satisfaction set of the CSL formula Φ at t_0, $\pi^{Z(t)}(s, Sat(\Phi, t_0))$ is a steady-state probability, $Prob^{Z(t)}(s, \phi, t_0)$ is defined as in Definition 4; and $Ind_{(s_j, t_0 \models \Phi)}$ is an indicator function, which shows whether a local state $s_j \in S$ satisfies formula Φ for a given overall state $\overline{x}$ at time t_0:

$$Ind_{(s_j, t_0 \models \Phi)} = \begin{cases} 1, & if \ \ s_j, t_0 \models \Phi, \\ 0, & if \ \ s_j, t_0 \not\models \Phi. \end{cases}$$

□

To check an MF-CSL formula at the global level (overall model), the local CSL formula has to be checked first, and the results are then used at the global level. The first step, namely CSL model-checking was explained in the previous sections, and for the algorithms for MF-CSL model-checking we refer to [25]. In the following we provide the example, which first shows the expressivity of the MF-CSL logic, and then provides the intuition behind the model-checking procedure.

Example 3. Let us consider the virus spread example to illustrate the expressive power of MF-CSL for mean-field models. In order to express the property that not more than 5% of the computers in the system are infected the following formula is used:

$$\mathbb{E}_{\leq 0{,}05} \text{ infected.}$$

The property "The percentage of all computers, which happen to have a probability lower than 10% of going from not infected to active infected state within 3 hours, is greater than 40%“ is expressed as

$$\mathbb{E}_{>0{,}4}(\mathcal{P}_{<0.1}(\text{not infected } U^{[0,3]} \text{ active})).$$

If one wants to ensure that the probability of a computer to be infected within two hours from now is less than 50%, the following property has to hold:

$$\mathbb{EP}_{<0.5}(\text{tt } U^{[0,2]} \text{ infected}).$$

Note that in the formula above the current state of the individual is not taken into account. If the percentage of not infected computers which will become infected within next two hours is of interest the formula has to be changed accordingly:

$$\Psi = \mathbb{EP}_{<0.5}(\text{not infected } U^{[0,2]} \text{ infected}).$$

If in a long run the system has to have a low probability (less then 2%) for a random computer to be infected the formula:

$$\mathbb{ES}_{<0.02} \text{ infected},$$

has to hold.

Let us consider the following MF-CSL formula:

$$\Psi = \mathbb{EP}_{<0.3}(\text{not infected } U^{[0,1]} \text{ infected}).$$

To check this formula against the occupancy vector $\overline{x}(0) = (0.8, 0.15, 0.05)$ we first have to check the CSL formula $\phi =$ (not infected $U^{[0,1]}$ infected), then we have to find the expected probability for the whole formula Ψ to hold according to the semantics of the MF-CSL, and, finally, compare it with the treashhold $p = 0.3$.

The probabilities $Prob^{Z(t)}(s, \text{not infected } U^{[0,1]} \text{ infected}, 0)$ that the underlying CSL formula holds for initial condition $\overline{x}(0) = (0.8, 0.15, 0.05)$ was found earlier in Section 5.5. It equals to 0.09, 0, and 0 for states s_1, s_2, and s_3, respectively.

According to Definition 6, the weighted sum of the entries of the occupancy vector $\overline{x}(0)$ and the respective probabilities in the local model define the expected probability $\mathbb{EP}(\phi)$:

$\sum_{j=1}^{K} x_j \cdot Prob^{Z(t)}(s_j, \phi, 0) = 0.8 \cdot 0.09 + 0.15 \cdot 0 + 0.05 \cdot 0 = 0.072 < 0.3.$

As one can see, the occupancy vector $\overline{x}(0) = (0.8, 0.15, 0.05)$ satisfies the MF-CSL formula $\mathbb{EP}_{<0.3}(\text{not infected } U^{[0,1]} \text{ infected})$.

In this section we provided the insides for both fluid model-checking and MF-CSL model-checking on the overall model. We showed how these two approaches are related and what kind of properties can be expressed and checked using both CSL and MF-CSL logics.

6 Conclusions

This paper illustrates several aspects of applying mean-field approximations for efficient analysis of large scale stochastic models. The purpose is to provide a self-contained, example-guided and accessible tutorial for researches that are interested in the area of mean-field.

The main idea of mean-field is to provide an approximation for a large number of interacting similar objects. In contrast to existing tutorials [9] this presentation starts from the single agent model and than abstracts to a large number of these objects using the mean-field, in addition, the single agent model within the whole population an inhomogeneous CTMC.

This paper features two case study, one on the analysis of Botnets, where indeed the distribution of objects is assumed to be uniform, and one on the analysis of gossip to show how the location of objects can be taken into account using spatial mean-field models.

The performance measures that are traditionally derived from such model are mainly steady-state and transient state distributions. However, exploiting the difference between the local object and the overall mean-field model allows to apply model checking techniques to derive more complex measures of interest. Section 5 repeats the main idea of fluid model checking, that can be used to check the single agent model and hints at a new logic, called MF-CSL that can be used to specify properties on the overall model. Note that we do not focus on all the details of these techniques, but aim to show how they can be used to analyse different aspects of the system.

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 [33] 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. To support the user in the correct application of these techniques, there are frameworks that allow for systematic application of mean-field techniques [9,21,36].

While the use of mean-field models in computer science already started in 1980 [28], still several open problems remain. For example, mean-field results are only reliable if the population is large enough, however it is still unclear whether and if so how this can be judged from the model at hand. Another interesting research topic would be to analyse the mean-field of models that include non-determinism.

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, 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 *QEST*, 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 *CONCUR*, volume 7454 of *LNCS*, pages 333–347. Springer, 2012.

9. L. Bortolussi, J. Hillston, D. Latella, and M. Massink. Continuous approximation of collective systems behaviour: A tutorial. *Performance Evaluation*, 70(5):317 – 349, 2013.
10. Cabspotting. http://stamen.com/clients/cabspotting.
11. A. Chaintreau, J.Y. Le Boudec, and N. Ristanovic. The age of gossip: spatial mean field regime. In *SIGMETRICS/Performance*, pages 109–120. ACM, 2009.
12. F. Ciocchetta and J. Hillston. Bio-pepa: A framework for the modelling and analysis of biological systems. *Theoretical Computer Science*, 410(33-34):3065–3084, 2009.
13. R.W.R. Darling and J.R. Norris. Differential equation approximations for Markov chains. *Probability Surveys*, 5:37–79, 2008.
14. 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. *IEEE Transactions on Software Engineering*, 28(10):956–969, 2002.
15. N. Gast and B. Gaujal. A mean field model of work stealing in large-scale systems. In *SIGMETRICS*, pages 13–24. ACM, 2010.
16. C.S. Gillespie. Moment closure approximations for mass-action models. *IET Systems Biology*, 3:52–58, 2009.
17. D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81(25):2340–2361, 1977.
18. 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, 2012.
19. R.A. Hayden and J.T. Bradley. A fluid analysis framework for a markovian process algebra. *Theoretical Computer Science*, 411(22-24):2260–2297, 2010.
20. J. Hillston. *A compositional approach to performance modelling.* Cambridge University Press, 1996.
21. J. Hillston. Fluid flow approximation of pepa models. In *QEST*, pages 33–43. IEEE Computer Society, 2005.
22. J. Hillston, M. Tribastone, and S. Gilmore. Stochastic process algebras: From individuals to populations. *The Computer Journal*, 2011.
23. L.P. Kadanoff. More is the Same; Phase Transitions and Mean Field Theories. *Journal of Statistical Physics*, 137:777–797, December 2009.
24. 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.
25. A. Kolesnichenko, P.T. de Boer, A.K.I. Remke, and B.R. Haverkort. A logic for model-checking mean-field models. In *DSN/PDF*, pages 1–12. IEEE Computer Society, 2013.
26. A. Kolesnichenko, A.K.I. 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 *EPEW*, volume 6977 of *LNCS*, pages 133–147. Springer, 2011.
27. T.G. Kurtz. Solutions of ordinary differential equations as limits of pure jump Markov processes. *Journal of Applied Probability*, 7(1):49–58, 1970.
28. T.G. Kurtz. *Approximation of population processes*, volume 36. Society for Industrial Mathematics, 1981.
29. D. Latella, M. Loreti, and M. Massink. On-the-fly Fast Mean-Field Model-Checking: Extended Version. Technical report, 2013.
30. J.Y. Le Boudec, D. McDonald, and J. Mundinger. A generic mean field convergence result for systems of interacting objects. In *QEST*, pages 3–18. IEEE Computer Society, 2007.

31. W.D. McComb. *Renormalization Methods: A Guide For Beginners.* OUP Oxford, 2004.
32. M. Mitzenmacher. The power of two choices in randomized load balancing. *IEEE Trans. Parallel Distrib. Syst.*, 12(10):1094–1104, October 2001.
33. A. Pourranjbar, J. Hillston, and L. Bortolussi. Dont Just Go with the Flow: Cautionary Tales of Fluid Flow Approximation. In *Computer Performance Engineering*, volume 7587 of *LNCS*, pages 156–171. Springer, 2013.
34. 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.
35. M. Tribastone. Relating layered queueing networks and process algebra models. In *WOSP/SIPEW*, pages 183–194, 2010.
36. M. Tribastone, S. Gilmore, and J. Hillston. Scalable differential analysis of process algebra models. *IEEE Trans. Software Eng.*, 38(1):205–219, 2012.
37. N.G. Van Kampen. *Stochastic Processes in Physics and Chemistry.* North-Holland Personal Library. Elsevier Science, 2011.
38. E. van Ruitenbeek and W.H. Sanders. Modeling peer-to-peer botnets. In *QEST*, pages 307–316. IEEE CS Press, 2008.
39. Wolfram Research, Inc. Mathematica tutorial. `http://reference.wolfram.com/mathematica/tutorial/IntroductionToManipulate.html`, 2010.