Mean-field Framework for Performance Evaluation of Push-Pull Gossip Protocols

Rena Bakhshi [a,*] Lucia Cloth [b] Wan Fokkink [a]
Boudewijn Haverkort [c,d]

[a] *Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands*

[b] *German University of Technology, Oman*

[c] *Centre for Telematics & Information Technology, University of Twente, Enschede, The Netherlands*

[d] *Embedded Systems Institute, Eindhoven, The Netherlands*

Abstract

Gossip protocols are designed to operate in very large, decentralised networks. A node in such a network bases its decision to interact (gossip) with another node on its partial view of the global system. Because of the size of these networks, analysis of gossip protocols is mostly done using simulations, but even they tend to be expensive in computation time and memory consumption.

We employ mean-field approximation for an analytical evaluation of gossip protocols. Nodes in the network are represented by small identical stochastic processes. Joining all nodes would result in an enormous stochastic process. If the number of nodes goes to infinity, however, mean-field analysis allows us to replace this intractably large stochastic process by a small deterministic process. This process approximates the behaviour of very large gossip networks, and can be evaluated using simple matrix-vector multiplications.

1 Introduction

We consider large-scale networks where a large number of nodes interact. In such networks, gossip protocols have shown to be a sensible paradigm for developing scalable and reliable communication mechanisms. For instance, information can

* Corresponding author
Email addresses: `rbakhshi@few.vu.nl` (Rena Bakhshi), `lucia.cloth@gutech.edu.om` (Lucia Cloth), `wanf@cs.vu.nl` (Wan Fokkink), `brh@cs.utwente.nl` (Boudewijn Haverkort).

be spread in a large-scale network if nodes periodically contact each other in a random fashion, and exchange their local information. The study of the emergent behaviour of gossip protocols demands the consideration of large-scale networks [30]. However, the large size of the network poses a serious challenge on modelling and analysis of a system of interacting nodes, even if the behaviour of a single node is relatively simple. Clearly, the analysis of gossip protocols with automated tools is hard – it is, for example, beyond the capabilities of current probabilistic model-checking tools [39]. Moreover, when a large number of nodes interact in a connected environment, various phenomena emerge that cannot be explained in terms of the behaviour of a single node. We are therefore interested in going from a detailed local model at node level to an abstract global model of the system.

In this paper, we resort to so-called *mean-field analysis*. The stochastic process representing the modelled system converges to a deterministic process if the size of the network tends to infinity (see, e.g., [20]). This convergence result provides an approximation for networks consisting of large numbers of interacting nodes that are symmetrically defined. The system is modelled on an abstract level as the distribution of nodes in the set of possible states; the mean-field method then allows us to observe the evolution of the distribution over time.

The notion of "mean-field" is often used in the literature, with different meanings. The mean-field concept was first introduced in statistical mechanics (e.g. [15]). It has been used in the context of Markov chain models of systems like plasma and dense gases where the strength of the interaction between particles is inversely proportional to the size of the system. A particle is seen as under a collective force generated by the other particles in a continuous time and space setting. Subsequently, the mean-field approximation enjoyed the attention of the neural networks community (e.g., [43]), and used in the area of chemical reaction networks (e.g., [38,41]). Furthermore, a number of deterministic continuous models including mean-field approximation have been considered also in the theory of epidemics; for more details see, e.g., [42,3,45,4].

In the area of communication networks, mean-field approximation have been applied in various forms to a variety of case studies, e.g. HTTP flows [7], TCP connections [8,6,5,47], bandwidth sharing [37], transportation networks [1], swarm robotic systems [40], reputation determination [20], queueing networks [23,35,49] and Internet congestion control [34].

We show that the mean-field method is well-suited for a performance evaluation of gossip protocols. Applying the method to these protocols is a natural choice, since gossip protocols are meant to operate in very large networks, consisting of interacting nodes that are symmetrically defined. In this paper, we present a modelling framework for push-pull gossip protocols; that is, where both nodes gossiping with each other share their local data.

Related work

In the context of gossip protocols, the mean-field modelling has been considered for analysis of the age of gossip [21], epidemic routing [51], and in the preliminary versions of this paper [10,11]. Several previous works on gossip protocols have used a notion of mean-value and infinite limit for the asymptotic analysis of algebraic gossip [46,24] and gossip-based membership protocols [18,19].

In particular, Bonnet [18] studied the evolution of the in-degree distribution of nodes executing the Cyclon protocol [48]. The states of the associated Markov chain represent the fraction of nodes with a specific in-degree distribution. The author showed that the system converges by constructing a generating function, a series whose coefficients encode the in-degree distribution. The generating function then enabled algebraic means to compute the mean value and the standard deviation of the stationary distribution. Stojanovic et al. [46] analysed and compared delay performance of network coding and cooperative diversity in a single-hop wireless network. The authors performed an asymptotic analysis (for the number of nodes $N \to \infty$) of the expected delay associated with the broadcasting of a file consisting of a certain number of packets.

In the recent work [12], the mean-field framework has been automated for large dynamic gossip networks since the calculation of a transition probability matrix is often a cumbersome and error-prone procedure. To represent topological information, the authors introduced a notion of classes in the framework. To represent dynamic networks, a notion of multiplicities in the state transitions has been introduced.

Contribution

This paper introduces a mean-field framework for evaluation of push-pull protocols. The paper extends the results previously published in [11,10] with the case study on a gossip-based information dissemination protocol, called the shuffling protocol. For the second case study on a gossip-based clock synchronization protocol GTP, we add extended details on the modelling.

Outline of the paper

This paper is further organised as follows. Sec. 2 gives a brief overview of the gossip paradigm, discusses different applications of gossip protocols, and explains two gossip protocols: an information dissemination protocol based on shuffling [27], and a basic version of a time protocol called GTP [29,28]. In Sec. 3, we describe the necessary mean-field theory, and devise a simple analytical model for gossip-based information dissemination as an illustrative example. This example demonstrates the mean-field theory for unidirectional communication models (called push-based or pull-based in terms of gossip protocols). Next, we show in Sec. 4 how pairwise node interaction can be modelled in the mean-field framework in the case of the shuffling protocol, using the mean-field convergence results from Sec. 3 and the state transition model of [14]. In Sec. 5 we present an analysis of basic version of GTP as a more complex application. Sec. 6 concludes our paper.

2 Gossip Protocols

Gossip-based protocols (sometimes referred to as epidemic protocols) are appealing in large-scale decentralised systems. In these protocols, nodes exchange data in a random fashion: a node chooses with some probability a peer to exchange information with. The gossip concept has originally been proposed for the analysis of database replication schemes [25].

2.1 A Generic Gossip Protocol

In gossip protocols, the information exchange between nodes can be implemented as one of the following policies: only the node that initiates a gossip sends local state data to its partner (push), a node-initiator requests state data from its gossip partner (pull), both nodes send their state information to each other (push-pull).

Figure 1 illustrates the skeleton of a generic push-pull gossip protocol. Each node has a local state s and executes two different threads, an active and a passive one. The active thread periodically initiates a state exchange with a random peer p by sending it a message containing the local state s, after which it waits for a response. The passive thread waits for a message sent by an initiator and replies to it with its local state. The random peer selection is based on the set of neighbours as determined by a membership protocol (e.g., [30]).

For a pair of nodes A and B, where A is the active node and B is the passive one, we describe the protocol from the point of view of each participating node. In particular, node A picks a neighbour B at random (method RandomPeer()) after a not necessarily constant (discrete) time span of length Δt, and initiates the state exchange (gossip) with it. It does so by send-

```
while true do
wait ($\Delta t$ time units)
$p \leftarrow$ RandomPeer();
prepare($s$);
send $s$ to $p$;
$s_p \leftarrow$ receive($\cdot$);
$s \leftarrow$ Update($s, s_p$);
```

(a) active thread

```
while true do
    $s_p \leftarrow$ receive($\cdot$);
    prepare($s$);
    send $s$ to sender($s_p$);
    $s \leftarrow$ Update($s, s_p$);
```

(b) passive thread

Fig. 1. A gossip protocol

ing (a part of) its local state s to B, and waits for B's response. Upon receipt of the response, node A updates its local state (according to the method $\mathsf{Update}(s, s_p)$). In response to being contacted by A, node B sends (part of) its local state to A and updates its local state accordingly (method $\mathsf{Update}(s, s_p)$).

2.2 Applications of Gossip

Method Update is protocol specific. It updates the local state of a node based on the previous local state, and the state information received from the random gossip partner. In gossip-based information dissemination protocols (as in, e.g., distributed news service protocols [31,27]), a finite list of data items (e.g., news items), called the cache, composes the local state of a node. The generic operation prepare(s) in Figure 1 is replaced by an operation $s \leftarrow \mathsf{RandomItems}()$. The method Update merges the list of old items with the list of received items. The measures of interest of these protocols include the number of copies of a data in the network after some time and the amount of time needed for the data to spread in the network.

In gossip-based membership management protocols, a finite set of peer addresses, called the partial view, comprises the local state of a node. The method Update (as in [48,2]) creates a new state through a sample of the union of the old and the received views. The performance metrics of these protocols include a distribution of the partial view size, and the number of nodes reached in the presence of node failures.

In probabilistic broadcasting (e.g., [50]), the state of a node is a flag that records whether the node is infected. Method Update sets the state to infected if the received state is infected. The performance of these protocols can be measured in, e.g., the time until all nodes have been infected.

In gossip-based distributed aggregation (e.g. [32]), the state of a node is a numeric value, which can be any parameter of the environment, such as a temperature or the current load. All values at nodes contribute to an aggregate value, computed using some aggregation function, for instance, average, sum, etc. The method Update simply returns the result of the aggregation function. For these protocols, a general measure of interest is the convergence of results of the aggregation function, but other measures depend on the aggregation function chosen.

We refer to [36] for a thorough survey on gossip applications.

2.3 The Shuffling Protocol

For the first case study, we describe the shuffling protocol, introduced in [27] and analysed in [14,13]. The protocol is an application of gossip-based information dissemination.

Every node maintains a local storage of the fixed size c for data items of general interest. Nodes disseminate n data items throughout the network by periodically gossiping with random neighbours and exchange s random items with each other. Upon receipt of the peer message, a node has to decide which items to keep, considering the limited capacity of its local storage. The decision is made according to the following steps:

- the node eliminates the received items that are already in its cache;
- the remaining received items are added into the local storage; if the total number of items exceeds the limit c, the node removes items among the ones that were sent to, but not received from the peer, until the storage has c items.

This procedure ensures that the items sent by the peer do not get lost, and, thus, the overall conservation of data in the network is ensured. The procedure is assumed to be atomic, that is, once a node initiates a gossip with another node, these two nodes cannot become involved in another interaction until the current contact is finished.

The state transition model

In [14], the protocol is modelled as a pairwise node interaction. The authors analyse the spread of a generic data item in the network. A state of the transition diagram in Fig. 2 is represented by a pair of bits, which indicate whether the data item is in the storage of the interacting nodes. The first bit of the state indicates the presence of the item in the storage of the active node A and the second one whether the item is in the storage of the contacted node B. Every transition between two states is labelled by a conditional probability $P(a_2b_2|a_1b_1)$, where (a_1, b_1) is the state before an exchange and (a_2, b_2) is the state after the exchange.

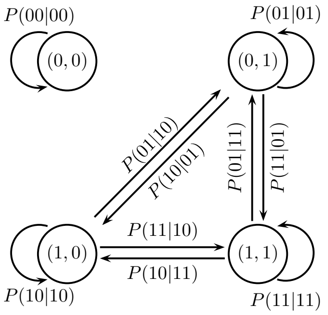


Fig. 2. State transition model

The state $(0, 0)$ in the model is disconnected from other states, because a node cannot discard or obtain the item, if its gossip partner does not have the item; this is not permitted by the protocol and, thus, the respective transition probabilities are equal to 0. All other states $(0, 1)$, $(1, 0)$, and $(1, 1)$ have three possible transitions each. For example, if two gossiping nodes are in the state $(1, 0)$, an active node can either: (1) pass on the item to its gossip partner with probability $P(01|10)$, or (2) decide not to gossip the item to its peer with probability $P(10|10)$, or (3) decide to gossip the item but keep its copy as well with probability $P(11|10)$.

These probabilities form a 4×4 transition matrix, calculated in [14]:

$$P(01|01) = P(10|10) = \tfrac{c-s}{c} \qquad P(01|11) = P(10|11) = \tfrac{s}{c}\tfrac{c-s}{c}\tfrac{n-c}{n-s}$$

$$P(10|01) = P(01|10) = \tfrac{s}{c}\tfrac{n-c}{n-s} \qquad P(11|11) = 1 - 2\tfrac{s}{c}\tfrac{c-s}{c}\tfrac{n-c}{n-s}$$

$$P(11|01) = P(11|10) = \tfrac{s}{c}\tfrac{c-s}{n-s} \qquad P(00|00) = 1$$

The transition probabilities are calculated under the assumption of a uniform distribution of data items over all nodes local storage. We refer to [14] for more details.

Differential equation models

These transition probabilities can be used to model two properties of the shuffling protocol, *replication* and *coverage*. Replication is defined as the fraction of nodes that possess a copy of the data item at a given time. Coverage is evaluated as the fraction of nodes that have seen the item within a given period.

Assuming $x(t)$ to be the fraction of nodes that hold the item at time t, the replication equation is:

$$\frac{dx}{dt} = (P(11|10) + P(11|01))\, x(t)(1 - x(t)) - (P(10|11) + P(01|11))\, x^2(t) \quad (1)$$

The first part of the equation covers the case that a gossiping node holds the item while the other does not, and both nodes hold the item after the exchange. The second part deals with the case that both gossiping nodes have the item, and one of them gives up its copy after the exchange.

Let $y(t)$ be the coverage of the item d at time t. The coverage equation is calculated as the fraction of nodes that do not hold the item and acquire it in one of the N gossip exchanges in the current round:

$$\frac{dy}{dt} = (1 - y(t)) \cdot \sum_{i=0}^{N-1} C(i) \cdot \Phi(i+1) \quad (2)$$

where $C(i)$ is the probability that a node is contacted exactly i times in a round:

$$C(i) = \binom{N-1}{i} \left(\frac{1}{N-1}\right)^{i} \left(\frac{N-2}{N-1}\right)^{(N-1)-i}$$

and $\Phi(i)$ the probability that once a node obtains the item after $m < i$ contacts, it does not lose the item in the remaining $i - m - 1$ contacts:

$$\Phi(i) = \sum_{m=0}^{i-1} (1 - \Phi(m)) \cdot P_{get} \cdot (P_{\neg lose})^{(i-m)-1}$$

In particular, the probability that an active node that does not have the item obtains it in a shuffle with probability

$$P_{get} = (P(10|01) + P(11|01)) \cdot x(t)$$

and a node that has the item does not lose it in a shuffle with probability

$$P_{\neg lose} = (P(10|10) + P(11|10)) \cdot (1 - x(t)) + (P(01|11) + P(11|11)) \cdot x(t)$$

Note that both equations are for a fully connected network. We refer to [14] for further details of these equations.

2.4 Gossiping Time Protocol

Protocols based on epidemic and gossip concepts have found various practical applications [36], including non-traditional gossip applications [22], such as gossip-based clock synchronisation. The Gossiping Time Protocol (GTP) [29,28] is a self-managing gossip time synchronisation protocol for peer-to-peer networks.

The protocol operates in a network of nodes, each equipped with a local clock, and assumes the presence of at least one node with accurate and robust time in the network. Time is disseminated throughout the network by letting nodes periodically gossip their clock samples. That is, each node periodically selects (initiates a gossip with) a random peer from the network to exchange time information with. The initiating period is determined by a value of the gossip delay parameter, which is the current delay between subsequent gossip interactions. The nodes subsequently exchange their local settings such that afterwards the node with the worse-quality time has adopted the higher-quality time of the other node. The protocol assumes a presence of the peer-sampling service [30], which allows a node to contact a uniformly randomly selected alive node.

In *basic* GTP, the quality of the time sample at a node is based on the distance from the time source to the node (hop count metric), that is, the number of nodes on the synchronisation path from the node to the time source. The time source has hop count equal to 0. Completely unsynchronised nodes have a hop count ∞. A gossiping node rejects the time sample if the hop count of its gossip partner is not smaller than its own. Furthermore, a node adopts a time sample if it has not been synchronised for a long time. That is, if the difference between the last update and the current time is larger than a timeout period, then a node accepts a time sample even though it may degrade its time quality (with respect to the hop count metric). Concisely, if the node decides to accept the sample, it synchronises its clock, and updates the values of the local variables. Namely, the node records the value of current time as the time of the last clock update, and sets the hop count to the value of the gossip partner hop count incremented by one. The GTP protocol parameters described above are stored as the following local variables, according to [28]: a gossip delay as `GOSSIPING_DELAY`, a time of the last clock update as `LAST_UPDATE`, a hop count as `TS_DISTANCE`, a timeout as `_STANDALONE_PERIOD_`.

Alternatively, each node may decide to adapt the rate at which it initiates a timestamp exchange (gossip frequency) based on its local settings. For instance, the better synchronised the node is, the lower the gossip frequency it may assume. In doing so, the gossip frequency gradually decreases when the network is synchronised and stable. Note that dynamic gossip frequency is beyond basic GTP, and is a simple optimization idea for the gradual version of GTP. The motivation behind it is that even though the timestamp exchange can be implemented to be inexpensive, it still consumes resources.

We show how a mean-field framework can be applied to a non-traditional application of gossip protocols, on the example of the basic GTP. That is, nodes execute basic GTP based on an immediate clock adjustment model, and change gossip frequencies, depending on a gossip delay. For the original protocol and its design details, we refer to [29,28].

3 Mean-field Modelling and Convergence

This section introduces the theory needed to apply mean-field results to gossip protocols. We stay close to the presentation in [20], but change notations when appropriate and simplify matters if possible in the gossip context.

3.1 Modelling and State Space

A *discrete-time Markov chain* (DTMC) is a stochastic process $\{Y(t) \mid t \in \mathbb{N}\}$ that takes values in a countable state space S. A DTMC obeys the Markov property, that is, the next state $j \in S$ is independent of the past, given the present state $i_l \in S$:

$$\Pr\{Y(t+1) = j \mid Y(0) = i_1, \ldots, Y(t) = i_t\} = \Pr\{Y(t+1) = j \mid Y(t) = i_t\}.$$

We consider a system of $N \in \mathbb{N}$ *interacting objects* that are identically defined. The object with index $n \in \{1, \ldots, N\}$ is represented by the discrete-time stochastic process $\{X_n^N(t) \mid t \in \mathbb{N}\}$ which takes values in the set $S = \{0, \ldots, K-1\}$ where $K = |S|$ is the number of different states.

Example 1 *In a gossip network, a node is represented by an interacting object. As a running example we consider a simple information dissemination protocol. A piece of information, e.g., the current time, is forwarded through the net. A node can be in one of two states: either it already has the information (state 0) or it is not yet informed (state 1). Hence, the state space for a node is* $S = \{0, 1\}$ *with* $|S| = K = 2$*. Let* m_0 *be a fraction of informed nodes, and* $p^N(m_0)$ *the probability of moving from state 1 to state 0. Figure 3 shows a graphical representation of the state-transition diagram describing such a node; the possible transitions and their probabilities will be explained later in this section.*

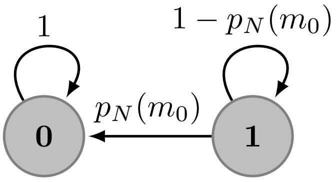


Fig. 3. A single node transition

The complete system is composed of the N objects and is, consequently, also described by a discrete-time stochastic process: $Y^N(t) = \left(X_1^N(t), \ldots, X_N^N(t)\right)$. Its state space is S^N which has $|S|^N = K^N$ elements. For the mean-field convergence result we assume that we can not distinguish objects that are in the same state. It then suffices to keep track of the fraction of objects in each state. These fractions are collected in another stochastic process $M^N(t) = (M_0(t), \ldots, M_{K-1}(t))$ called the *occupancy measure*. Its elements are defined as

$$M_i^N(t) = \frac{1}{N} \sum_{n=1}^{N} 1_{\{X_n^N(t)=i\}}, \; i \in S,$$

where $1_{\{X_n^N(t)=i\}} = 1$ if $X_n^N(t) = i$, and 0 otherwise. Its state space $S_M^N \subset \mathbb{R}^K$ has

$$\left|S_M^N\right| = \binom{K+N-1}{K-1}$$

elements (the number of ways to distribute N objects over the K states they can be in). One state from this state space is denoted $\mathbf{m} = (m_0, m_1, \ldots, m_{K-1}) \in S_M^N$, where m_i is the fraction of nodes in the state i.

Example 2 *For the information dissemination example, the state space of the occupancy measure is*

$$S_M^N = \left\{ \left(\frac{k}{N}, 1 - \frac{k}{N} \right) \middle| \, k \in \{0, \ldots, N\} \right\}.$$

Its size is

$$\left|S_M^N\right| = \binom{2+N-1}{2-1} = N + 1.$$

Normally, the choice of the occupancy measuredepends on the local parameters of a node, which, in its turn, depends on the application of gossip.

3.2 Local Transition Probabilities

The evolution of the system of interacting objects is described by the *local* transition probabilities of each object. The next state of any object not only depends on the current state of the object *but also* on the current occupancy measure $\mathbf{m}$:

$$P_{i,j}^N(\mathbf{m}) = \Pr\{X_n^N(t+1) = j \mid X_n^N(t) = i, M^N(t) = \mathbf{m}\}, i, j \in S,\ \mathbf{m} \in S_M^N \quad (3)$$

These probabilities are the same for all objects. They are gathered into the transition probability matrix $P^N(\mathbf{m})$. These local transition probabilities determine the unique transition probability matrix for the global system $Y^N(t)$, which is a DTMC because its next state (= occupancy measure) only depends on the current state.

In contrast to Example 1, the exchange of information between nodes can be bidirectional as, for instance, in case of push-pull based gossip protocols. In these gossip protocols, both interacting nodes update their local states depending on both of their current states.

Formally, two communicating objects (n_1, n_2) undergo a state transition according to

$$\begin{aligned} P^N_{(i_1,j_1),(i_2,j_2)}(\mathbf{m}) = \Pr\{X^N_{n_1}(t+1) = j_1, X^N_{n_2}(t+1) = j_2 \mid \\ X^N_{n_1}(t) = i_1, X^N_{n_2}(t) = i_2, M^N(t) = \mathbf{m}\}, \quad (4) \end{aligned}$$

where $i_1, i_2, j_1, j_2 \in S$, and $\mathbf{m} \in S_M^N$. Analogous observations regarding pairwise interaction of objects in the mean-field models have been made in [16]. In the pairwise interaction model, each transition probability $P^N_{(i_1,j_1),(i_2,j_2)}(\mathbf{m})$ takes into account current occupancy measures $M_{i_1}(t)$ and $M_{i_2}(t)$ of both states i_1 and i_2. Thus, at every time step t the value of the probability $P^N_{(i_1,j_1),(i_2,j_2)}(\mathbf{m})$ is recomputed based on the current value of the occupancy measure $M_{i_2}(t)$.

In the next sections, we model pairwise node interaction and pairwise state update of the nodes. Using the pairwise interaction model (4) in the mean-field framework, we analyse the push-pull information dissemination protocol in Sec. 4 and a basic version of GTP in Sec. 5. In this section, however, we keep to the pull model of a node, introduced in Example 1, and the transition equation (3).

Example 3 *A node can only move from being uninformed (state 1) to being informed (state 0). Afterwards it stays in state 0 forever, that is, it never forgets. Suppose that in each time step a node A initiates a gossip interaction with probability g. It randomly chooses a partner node B among the $N-1$ other nodes. If B is already* *informed and A is not, A moves to state 0, so that we model a simple pull protocol. Note that m_0 is the fraction of informed nodes in the system and $m_1 = 1 - m_0$ the fraction of uninformed nodes. The total probability for moving from state 1 to state 0 equals*

$$p^N(m_0) = P^N_{1,0}((m_0, m_1)) = g \cdot \frac{m_0 \cdot N}{N-1}.$$

Here, $m_0 \cdot N$ is the number of informed nodes and $m_0 \cdot N/(N-1)$ is the probability that a node chooses an informed node out of the $N-1$ possible nodes (it does not pick itself) as gossip partner. The complete probability matrix is then given by

$$P^N((m_0, m_1)) = \begin{pmatrix} 1 & 0 \\ p^N(m_0) & 1 - p^N(m_0) \end{pmatrix}.$$

For the global system, the probability to move from m_0 informed nodes to m_0' informed nodes, for $m_0' \geq m_0$, equals

$$\binom{m_1 \cdot N}{(m_0' - m_0) \cdot N} \left(p^N(m_0)\right)^{(m_0' - m_0)N} \left(1 - p^N(m_0)\right)^{m_1' N},$$

where $m_1 = 1 - m_0$, $m_1' = 1 - m_0'$. This binomial expression is composed of the number of possibilities to choose exactly the "missing" $(m_0' - m_0) \cdot N$ objects out of the $m_1 \cdot N$ uninformed nodes; these then all have to take the transition to state 0, and all other $m_1' \cdot N$ nodes remain in state 1.

Consider now the occupancy measure $M^N(t)$ of the system at a given time $t \in \mathbb{N}$. Recall that $M^N(t)$ is a random variable. For a given initial occupancy measure $\mathbf{m}_0^N$, there are two ways to determine the distribution of $M^N(t)$: first, we can calculate the transient distribution analytically at time t, requiring t vector-matrix multiplications with a vector of size $|S_M^N|$. Second, we can employ discrete-event simulation to estimate the distribution. Often only discrete-event simulation is possible since, for large N, the size of the state space makes the analytical computation of the transient probabilities practically infeasible. But even discrete-event simulation of this large DTMC is expensive.

Suppose the number of local states of nodes is s. Then the time complexity per step in the mean-field method is s^2, i.e. the cost of matrix-vector multiplication. For our simple example, the time complexity is only $2 \cdot 2 = 4$ per time step. Already a single Monte-Carlo simulation of the system with only 100 nodes is much more expensive. Moreover, for a reasonable confidence interval one needs to repeat the simulations for at least 100 times. For more complex systems, in general, to observe the emergent behaviour may require several thousands of nodes and thousands of runs of the simulations, whereas the mean-field method requires only one run (independent of the number of nodes), providing the limiting behaviour of the system.

Roughly speaking, the complexity is s^2 for the mean-field method compared to $f(N) \cdot r$ of the Monte-Carlo simulations, where $f(N)$ is a cost for simulating one run, which is at least linear with respect to the number of nodes N, and r is the number of runs. Note that $f(N)$ can be large; for example, the Monte-Carlo simulations of the shuffling protocol for a single pairwise communication costs $c \cdot \log c$ (see [14] for details), where c is the local storage size of a node.

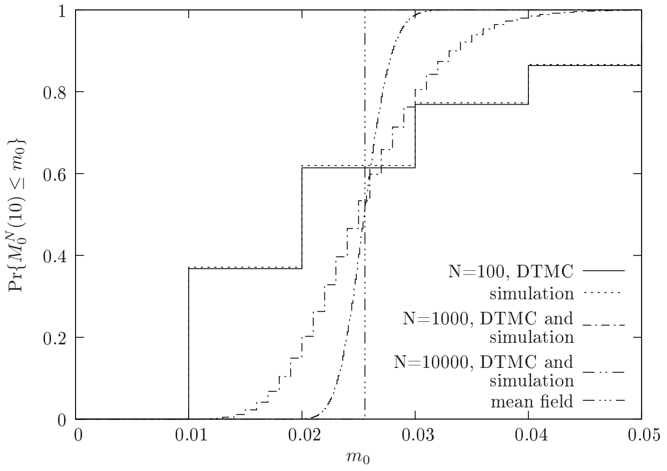


Fig. 4. Distribution of $M_0^N(10)$ for $g = 0.1$ and $M^N(0) = (0.01, 0.99)$

Example 4 *Figure 4 shows the analytically computed distribution of the fraction of informed nodes at time* $t = 10$*, for gossip probability* $g = 0.1$ *and initial occupancy measure* $M^N(0) = (0.01, 0.99)$*. The analytical results in the figure are obtained using mean-field analysis (the vertical line at* 0.025*), and by using matrix-vector multiplications with a vector size* $|S_M^N|$*, as described above. Note that the distribution is "more deterministic" for larger* N*.*

We also simulated this simple dissemination protocol in a round-based fashion similar to simulations in PeerSim [33]. In one round, which equals one time step, each uninformed node gossips with probability g *and picks a random peer. If this peer is already informed, the number of informed nodes for the next round is increased by one. Using 10000 independent runs for each curve, the resulting distributions for* $M_0^N(10)$ *are also shown in Figure 4.*

3.3 Convergence Result

At this point, the so-called mean-field convergence result applies. It captures the limiting behaviour of the complete system if the number of objects N goes to infinity and so provides an approximation for the occupancy measure for large N. The requirement is that for all local states $i, j \in S$, all $\mathbf{m} \in \mathbb{R}^K$ and $N \to \infty$:

$P^N_{i,j}(\mathbf{m})$ converges uniformly[1] in $\mathbf{m}$ to some $P_{i,j}(\mathbf{m})$, which is a continuous function of $\mathbf{m}$.

If this requirement is satisfied, the occupancy measure converges almost surely to a deterministic limit. This means that in case $N \to \infty$, for each local state i the fraction $M^N_i(t)$ of objects in state i at time t is known with probability one.

Theorem 1 (cf. [26]) *Fix the initial occupancy measure to be identical for all $N \in \mathbb{N}$: $M^N(0) = \mu(0)$. Define the limit of the local probability matrix:*

$$P(\mathbf{m}) = \lim_{N\to\infty} P^N(\mathbf{m}),\ \mathbf{m} \in \mathbb{R}^K.$$

Define the deterministic process $\mu(t+1) = \mu(t) \cdot P(\mu(t))$.

Then for any $t \in \mathbb{N}$, $\lim\limits_{N\to\infty} M^N(t) = \mu(t)$, with probability 1, that is, $\mu(t)$ is the deterministic limit occupancy measure for $N \to \infty$.

For large N we can now approximate the stochastic process for the occupancy measure by this deterministic process.

Example 5 *The limit of the probability to move from state 1 to state 0 is*

$$p(m_0) = \lim_{N\to\infty} g \cdot \frac{m_0 \cdot N}{N-1} = g \cdot m_0,$$

which is continuous in m_0. The requirement for the application of the mean-field convergence result is thus satisfied. If we set $\mu(0) = (0.01, 0.99)$ and $g = 0.1$, the deterministic limit for time $t = 10$ is $\mu(10) = (0.0256, 0.9744)$, which is computed by ten matrix-vector multiplications. It is indicated by the vertical line for m_0 in Figure 4.

3.4 A Methodology for the Mean-field Analysis of Gossip Protocols

We summarise how mean-field analysis can be used for the performance evaluation of gossiping protocols. Our methodology consists of the following steps:

Step 1 – System description The specification of a system helps to obtain not only a better (more modular) description, but also a clear understanding and an abstract view of the system. In general, it is hard to give a full specification of a system or protocol under study. Such a study is usually done on a simplified system model of the actual protocol: one has to decide which characteristics of

[1] A sequence f_N of real valued functions converges uniformly with limit f if for every $\varepsilon > 0$ there exists a natural number n such that for all x and all $N \geq n$ we have $|f_N(x) - f(x)| < \varepsilon$.

the protocol should be studied, and which parameters of the protocol should be modelled in order to study these characteristics. In order to simplify the system model, assumptions should be made. These assumptions should be supported by experimental study.

Step 2 – Identification of local states and transitions This step requires identification the set S of local states of a node. The states should reflect all relevant situations a node can be in. Transitions between local states usually occur because of gossip interactions.

Step 3 – Transition probabilities The (local) transition probabilities depend on the global state of the gossip network model. The probabilities have to be investigated thoroughly. A node might also behave intrinsically in a probabilistic way. At the end of this step stands a directive of how to compute the transition probability matrix depending on the current global state.

Step 4 – Mean-field convergence requirements Only if the local transition probabilities converge appropriately for $N \to \infty$, can we successfully apply the mean-field convergence theorem.

Step 5 – Mean-field limit Finally, we can compute the mean-field limit for our model using Theorem 1. With the obtained results we can test and compare different designs.

4 A Mean-field Model for the Shuffling Protocol

We apply mean-field theory to the shuffling protocol, using the analytical model of [14]. The specification of the shuffling protocol in Sec. 2.3 and [27,14] corresponds to Step 1 in our mean-field framework. Steps 2 and 3 are covered by the following subsections (4.1–4.3). Finally, Sec. 4.4 covers Steps 4 and 5.

We demonstrate how the transition diagram model, described in Sec. 2.3 for pairwise node interaction, can be used for our mean-field framework.

4.1 State Space

We use the abstraction of the shuffling protocol from [14] and analyse the spread of a distinguished item in the network. The state of a node in the shuffling protocol is defined by a pair (g, d). The first component g denotes the gossip delay. It defines the speed with which nodes communicate with each other. When the gossip delay reaches zero a node initiates a shuffle. The second element d is a two-valued integer (bit) indicating whether a node possesses the item.

The state space of a node is $S = \{0, \ldots, G_{\max}\} \times \{0, 1\}$, and the size of the state space is $|S| = (G_{\max}+1) \cdot 2$, where $G_{\max}$ is the maximal gossip delay. Note that we consider a discrete time model, that is, the system proceeds in discrete time steps $t \in \mathbb{N}$.

4.2 Local Transition Probabilities

The behaviour of a node is based on its state and the current occupancy measure $\mathbf{m}$. The expression $(g, d \mid g > 0)$ denotes any state where $g > 0$ and d is chosen arbitrarily from the set $\{0, 1\}$. The expression $\mathbf{m}_{(g,d|g\leq g')}$ is an example for the abbreviation of a sum of occupancy fractions, calculated as:

$$\mathbf{m}_{(g,d|g\leq g')} = \sum_{g=0}^{g'} \mathbf{m}_{(g,0)} + \sum_{g=0}^{g'} \mathbf{m}_{(g,1)}.$$

The transition probability $P^N_{(g,0|g>0),(g-1,0)}(\mathbf{m})$ denotes the transition probability from the state $(g, 0)$ to the state $(g-1, 0)$ for any $g > 0$.

Figure 5 depicts the two-dimensional diagram for a single node transition. According to our model in Sec. 4.1, each possible state of the node is defined by its gossip delay and by the presence of the item at the node. The x-axis in Figure 5 corresponds to the value of the gossip delay g, and the y-axis to the value of d. For all transitions represented by the arrow, we calculate their respective state transition probabilities.

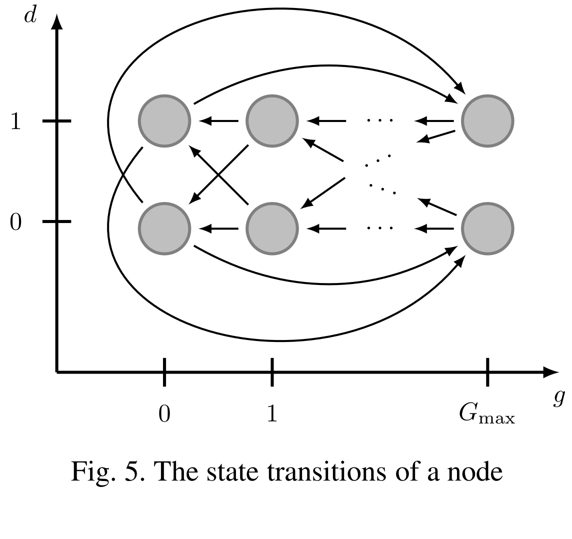


Fig. 5. The state transitions of a node

Active nodes

Periodically, a node A initiates the interaction with a random peer B. The period is governed by the gossip delay g, and the length is equal to $G_{\max}$. Thus, whenever g of a node A reaches 0, the node becomes *active* and initiates the interaction with a peer B, selected at random among $N-1$ nodes (excluding A itself).

As described in Sec. 2.3, two gossiping nodes cannot become involved in another interaction until the current contact is finished. We address this assumption by introducing the notion of a *collision*. Namely, once active node A randomly selected B as its gossip partner, the probability that no collision occurs, $\text{noc}^N(\mathbf{m})$, ensures that other active nodes select neither A nor B. Thus, the probability that no collision occurs is equal to:

$$\text{noc}^N(\mathbf{m}) = \left(\frac{N-3}{N-1}\right)^{\mathbf{m}_{(0,d)}\cdot N-1}.$$

That is, all active nodes $\mathbf{m}_{(0,d)} \cdot N$ excluding A choose one of the remaining $N-3$ nodes as their gossip partner, excluding A, B and the node itself.

We calculate the transition probabilities $P^N_{(0,d),(G_{\max},d')}(\mathbf{m})$ for all combinations of $d, d' \in \{0,1\}$. An active node is in one of two possible states: (a) the node holds the item (i.e. state $(0,0)$), or (b) the node does not have the item (i.e. state $(0,1)$). In either case, the node will either not have the item after the shuffle or obtain the item during the exchange. That is, the new state after the interaction is either $(G_{\max}, 0)$ or $(G_{\max}, 1)$.

If an active node does not hold the item before the contact, the value of d remains 0 in the following three cases: (1) collision occurred while talking to a passive node, (2) the chosen peer is active itself, or (3) independent of the peer state, a node did not obtain the item after the contact (by the probabilities $P(00|00)$ and $P(01|01)$).

$$\begin{aligned} P^N_{(0,0),(G_{\max},0)}(\mathbf{m}) &= \sum_{i=0}^{1} \frac{\mathbf{m}_{(g',i|g'>0)}\cdot N}{N-1} \cdot (1-\text{noc}^N(\mathbf{m})) \\ &+ \sum_{i=0}^{1} \frac{\mathbf{m}_{(0,i)}\cdot N - 1 + i}{N-1} \\ &+ \sum_{i=0}^{1} \frac{\mathbf{m}_{(g',i|g'>0)}\cdot N}{N-1} \cdot P(0i|0i)\cdot \text{noc}^N(\mathbf{m}) \end{aligned} \tag{5}$$

First, an active node selects a passive node with probability $\mathbf{m}_{(g',i|g'>0)} \cdot N/(N-1)$, but the interaction fails due to a collision that occurs with probability $1-\text{noc}^N(\mathbf{m})$. Second, the active node chooses another active node with probability $(\mathbf{m}_{(0,0|g'>0)} \cdot (N-1)+\mathbf{m}_{(0,1|g'>0)} \cdot N)/(N-1)$, and thus, the contact fails. Third, the active node chooses a passive node with probability $\mathbf{m}_{(g',i|g'>0)} \cdot N/(N-1)$ and successfully communicates with the probability $\text{noc}^N(\mathbf{m})$. The node does not get the item with probability $P(00|00)$, if the peer did not have the item as well, and with $P(01|01)$, if the peer had the item.

The item is obtained after the interaction, if a passive peer holds the item before the interaction, according to the probabilities $P(11|01)$ and $P(10|01)$:

$$P^N_{(0,0),(G_{\max},1)}(\mathbf{m}) = \frac{\mathbf{m}_{(g',1|g'>0)} \cdot N}{N-1} \cdot \sum_{j=0}^{1} P(1j|01) \cdot \text{noc}^N(\mathbf{m}) \tag{6}$$

That is, an active node chooses a passive peer that had the item with probability $\mathbf{m}_{(g',1|g'>0)} \cdot N/(N-1)$, and after the communication without collision $\text{noc}^N(\mathbf{m})$, the active node obtains the item with probability $P(11|01) + P(10|01)$.

If the active node holds the item before interaction, the value of d stays as 1, because (1) a collision occurred while talking to passive nodes, (2) the chosen peer is an active node, or (3) the item stays in the local cache, independent of whether the peer possesses the item before or after the shuffle, determined by the probabilities $P(10|10)$, $P(11|10)$, $P(10|11)$ and $P(11|11)$.

$$\begin{aligned} P^N_{(0,1),(G_{\max},1)}(\mathbf{m}) = & \sum_{i=0}^{1} \frac{\mathbf{m}_{(g',i|g>0)} \cdot N}{N-1} \cdot (1 - \text{noc}^N(\mathbf{m})) \\ & + \sum_{i=0}^{1} \frac{\mathbf{m}_{(0,i)} \cdot N}{N-1} \\ & + \sum_{i=0}^{1} \frac{\mathbf{m}_{(g',i|g'>0)} \cdot N - i}{N-1} \cdot \sum_{j=0}^{1} P(1j|1i) \cdot \text{noc}^N(\mathbf{m}) \end{aligned} \tag{7}$$

However, the item is given away by an active node after the exchange with probability

$$P^N_{(0,1),(G_{\max},0)}(\mathbf{m}) = \sum_{i=0}^{1} \frac{\mathbf{m}_{(g',i|g'>0)} \cdot N}{N-1} \cdot P(01|1i) \cdot \text{noc}^N(\mathbf{m}) \tag{8}$$

since, either (1) the active node chooses a passive peer that did not have the item with the probability $\mathbf{m}_{(g',0|g'>0)} \cdot N/(N-1)$, successfully communicated with probability $\text{noc}^N(\mathbf{m})$, and passed the item to the peer with probability $P(01|10)$; or, (2) the active node chooses a passive peer that had the item with probability $\mathbf{m}_{(g',1|g'>0)} \cdot N/(N-1)$, successfully communicated with probability $\text{noc}^N(\mathbf{m})$, and dropped its copy of the item with probability $P(01|11)$.

Passive nodes

A passive node, that is, a node with $g > 0$, can be contacted by an active peer, resulting in an update of its state. In all cases, the gossip delay is decreased by one, counting down to the next gossip initiation. Similar to active nodes, we distinguish four main transitions, depending on whether the item is at the node before or after gossiping.

The probability that a passive node does not obtain the item after the gossip is equal to the following sum:

$$\begin{aligned} P^N_{(g,0|g>0),(g-1,0)}(\mathbf{m}) = &\sum_{i=0}^{1} \frac{\mathbf{m}_{(0,i)} \cdot N}{N-1} \cdot (1 - \mathrm{noc}^N(\mathbf{m})) \\ &+ \left(1 - \sum_{i=0}^{1} \frac{\mathbf{m}_{(0,i)} \cdot N}{N-1}\right) \\ &+ \sum_{i=0}^{1} \frac{\mathbf{m}_{(0,i)} \cdot N}{N-1} \cdot P(i0|i0) \cdot \mathrm{noc}^N(\mathbf{m}) \end{aligned} \tag{9}$$

The sum consists of the three components: (1) the probability $\mathbf{m}_{(0,i)} \cdot N/(N-1)$ that the node is chosen by an active node, but a collision occurred with probability $1 - \mathrm{noc}^N(\mathbf{m})$; (2) the probability that a node is not chosen as a peer by any active node; (3) the probability $\mathbf{m}_{(0,i)} \cdot N/(N-1)$ to be chosen by an active node, and after successful communication (with probability $\mathrm{noc}^N(\mathbf{m})$), not to obtain the item with probability $P(00|00)$, if the active node did not have the item, and with $P(10|10)$, otherwise.

A passive node obtains the item with probability

$$P^N_{(g,0|g>0),(g-1,1)}(\mathbf{m}) = \frac{\mathbf{m}_{(0,1)} \cdot N}{N-1} \cdot \sum_{j=0}^{1} P(j1|10) \cdot \mathrm{noc}^N(\mathbf{m}) \tag{10}$$

that is, the probability that the node is chosen by an active peer that holds the item $\mathbf{m}_{(0,1)} \cdot N/(N-1)$, successfully communicates according to the probability $\mathrm{noc}^N(\mathbf{m})$, and the node obtains the item with probability $P(01|10) + P(11|10)$.

The remaining two probabilities cover the case when a passive node has the item before the interaction. It can either keep the item or give it away as the result of an interaction with an active peer with the following probabilities.

The probability that a passive node keeps the item after the interaction is computed as the sum

$$\begin{aligned} P^N_{(g,1|g>0),(g-1,1)}(\mathbf{m}) = &\sum_{i=0}^{1} \frac{\mathbf{m}_{(0,i)} \cdot N}{N-1} \cdot (1 - \mathrm{noc}^N(\mathbf{m})) \\ &+ \left(1 - \sum_{i=0}^{1} \frac{\mathbf{m}_{(0,i)} \cdot N}{N-1}\right) \\ &+ \sum_{i=0}^{1} \frac{\mathbf{m}_{(0,i)} \cdot N}{N-1} \cdot \sum_{j=0}^{1} P(j1|i1) \cdot \mathrm{noc}^N(\mathbf{m}) \end{aligned} \tag{11}$$

First, the node is chosen by an active node with probability $m_{(0,i)} \cdot N/(N-1)$, but a collision occurred during the interaction with probability $1 - \mathrm{noc}^N(\mathbf{m})$. The second summand expresses the probability that the passive node is not selected by any active node as a peer. Third, after the successful communication with an active peer with probability $m_{(0,i)} \cdot N/(N-1) \cdot \mathrm{noc}^N(\mathbf{m})$, the passive node keeps the item with probability $P(01|i1) + P(11|i1)$, where i indicates whether the active peer has the item or not.

The probability that a node gives away its copy of the item is calculated as:

$$P^N_{(g,1|g>0),(g-1,0)}(\mathbf{m}) = \sum_{i=0}^{1} \frac{\mathbf{m}_{(0,i)} \cdot N}{N-1} \cdot P(10|i1) \cdot \mathrm{noc}^N(\mathbf{m}) \tag{12}$$

Namely, after the successful communication with an active peer with probability $m_{(0,i)} \cdot N/(N-1) \cdot \mathrm{noc}^N(\mathbf{m})$, the passive node drops its copy of the item with probability $P(10|01)$, if the peer did not have the item, and $P(10|11)$, otherwise.

4.3 Coverage Property of the Protocol

The model, described in Sec. 4.1-4.2, is sufficient to perform the analysis of the shuffling protocol based on the *replication* property. Recall that replication expresses the fraction of nodes that possess a copy of the data item at a given time. Other properties of the shuffling protocol include *coverage*, i.e., the fraction of nodes that have seen the item within the given period.

In order to analyse the performance of the protocol according to coverage, we need to extend the state space with an additional parameter, $o \in \{0,1\}$. The parameter $o = 0$, if a node has not held the item previously, and $o = 1$, otherwise. Once the item is acquired in one of the exchanges, o stays equal to 1. Thus, the state of a node is a triple (g, d, o). The state space of the extended model is $S = \{0, \ldots, G_{\max}\} \times \{0,1\} \times \{0,1\}$ with the size $|S| = (G_{\max} + 1) \cdot 4$.

The transition probabilities for the replication model have to be adapted according to the new state space. Clearly, if a node holds the item ($d = 1$), the node becomes informed about the item ($o = 1$). Moreover, once a node has obtained the item, it cannot 'forget' even after losing the copy during one of the exchanges:

$$P^N_{(g,d,1|g>0),(g-1,d',0)}(\mathbf{m}) = P^N_{(0,d,1),(G_{\max},d',0)}(\mathbf{m}) = 0$$

Thus, several transition probabilities are computed in a similar way. The transition probabilities for active nodes from (5), (8), and (7) remain:

$$\begin{aligned} P^N_{(0,0,o),(G_{\max},0,o)}(\mathbf{m}) &= P^N_{(0,0),(G_{\max},0)}(\mathbf{m}), \quad o \in \{0,1\} \\ P^N_{(0,1,1),(G_{\max},0,1)}(\mathbf{m}) &= P^N_{(0,1),(G_{\max},0)}(\mathbf{m}) \end{aligned}$$

$$P^N_{(0,1,1),(G_{\max},1,1)}(\mathbf{m}) = P^N_{(0,1),(G_{\max},1)}(\mathbf{m})$$

All four probabilities cover the transitions, where a value of the parameter o does not change. Likewise, the corresponding transition probabilities for passive nodes from (9), (12), and (11) remain:

$$\begin{aligned} P^N_{(g,0,o|g>0),(g-1,0,o)}(\mathbf{m}) &= P^N_{(g,0|g>0),(g-1,0)}(\mathbf{m}), \quad o \in \{0,1\} \\ P^N_{(g,1,1|g>0),(g-1,1,1)}(\mathbf{m}) &= P^N_{(g,1|g>0),(g-1,1)}(\mathbf{m}) \\ P^N_{(g,1,1|g>0),(g-1,0,1)}(\mathbf{m}) &= P^N_{(g,1|g>0),(g-1,0)}(\mathbf{m}) \end{aligned}$$

We distinguish the transitions in which a node acquires the item for the first time, and in which a node that held the item previously obtains it again. Although we differentiate these cases, the transition probabilities are equal to the probability that a node obtains a copy of the item of (10) and (6):

$$\begin{gathered} P^N_{(g,0,0|g>0),(g-1,1,1)}(\mathbf{m}) = P^N_{(g,0,1|g>0),(g-1,1,1)}(\mathbf{m}) = P^N_{(g,0|g>0),(g-1,1)}(\mathbf{m}) \\ P^N_{(0,0,0),(G_{\max},1,1)}(\mathbf{m}) = P^N_{(0,0,1),(G_{\max},1,1)}(\mathbf{m}) = P^N_{(0,0),(G_{\max},1)}(\mathbf{m}) \end{gathered}$$

Note that in all probabilities the occupancy measure $\mathbf{m}_{(g,0)}$ for any g is now treated as the sum $\mathbf{m}_{(g,0,0)} + \mathbf{m}_{(g,0,1)}$, and $\mathbf{m}_{(g,1)}$ for any g simply becomes $\mathbf{m}_{(g,1,1)}$. The same applies to the probability that no collision occurs:

$$\text{noc}^N(\mathbf{m}) = \left(\frac{N-3}{N-1}\right)^{(\mathbf{m}_{(0,0,0)}+\mathbf{m}_{(0,0,1)}+\mathbf{m}_{(0,1,1)})\cdot N-1}.$$

That is, given two nodes interacting with each other, all other active nodes choose peers different from the gossiping pair.

4.4 Mean-field Limits

In the limit, the value of the probability $\text{noc}^N(\mathbf{m})$ for the replication model is

$$\text{noc}(\mathbf{m}) = \lim_{N\to\infty} \text{noc}^N(\mathbf{m}) = \lim_{N\to\infty} \left(\frac{N-3}{N-1}\right)^{\mathbf{m}_{(0,d)}\cdot N-1} = e^{-2\cdot\mathbf{m}_{(0,d)}},$$

where $\mathbf{m}_{(0,d)}$ denotes the sum $\mathbf{m}_{(0,0)} + \mathbf{m}_{(0,1)}$, and for the coverage model is:

$$\text{noc}(\mathbf{m}) = \lim_{N\to\infty} \text{noc}^N(\mathbf{m}) = \lim_{N\to\infty} \left(\frac{N-3}{N-1}\right)^{\mathbf{m}_{(0,d,o)}\cdot N-1} = e^{-2\cdot\mathbf{m}_{(0,d,o)}},$$

where $\mathbf{m}_{(0,d,o)}$ is the sum $\mathbf{m}_{(0,0,0)} + \mathbf{m}_{(0,0,1)} + \mathbf{m}_{(0,1,1)}$. We note that in the transition probabilities for active and passive nodes calculated above, factors like $N/(N-1)$ converge to 1 if $N \to \infty$.

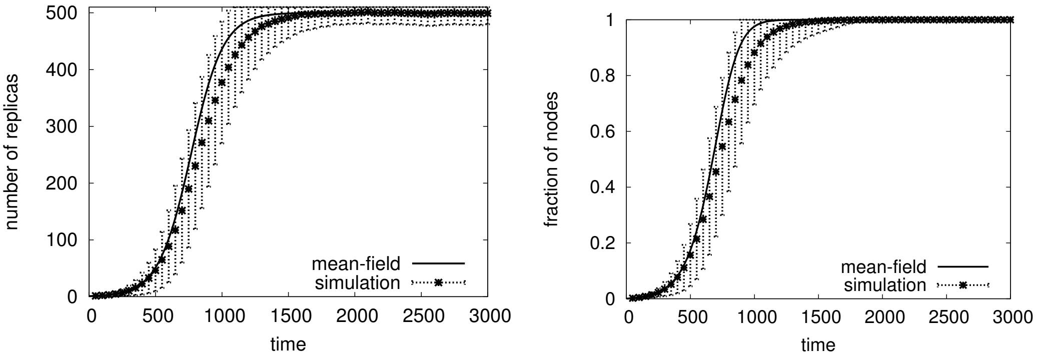
4.5 Comparison with Previous Results

We compare the results of our mean-field model with the results of the properties observed while running the actual protocol in a large-scale deployment. In case of the shuffling protocol, we simulate the network of 2500 nodes on a single workstation, using a Java-based implementation. Each node has local storage size $c = 100$, and exchange message size $s = 50$. All nodes are randomly divided into 10 different groups with the different values of gossip delay. The period between two consecutive contact initiations is fixed and set to 10. The nodes from the group with $g = 0$ are active nodes, that initiate simultaneously an exchange with the peers chosen uniformly at random. If a node contacts two gossiping nodes, the interaction between all three nodes fails. Initially a new item is introduced at one node in a network, in which $n = 500$ items are uniformly distributed over the local storage of all nodes. After each step, we measure the total number of copies of the distinguished item in the network, and the number of nodes that have seen this item. Each simulation curve in Figure 6 represents the average calculated over 500 runs.

Relating our mean-field model to these settings, we set the fraction of nodes that possess the item initially to 1/2500. The maximum gossip delay $G_{\max} = 9$. All nodes have a gossip delay uniformly distributed between 0 and $G_{\max}$. Each discrete time step, we record the values of the occupancy measure $\mathbf{m}$.

Figure 6(a) shows the evolution of the number of copies of the distinguished item in the network over time. For the mean-field model we have multiplied the fraction of nodes with $d = 1$ by 2500 to compare with the simulation curve. Both curves of simulation and mean-field model are quite close, with a small difference between steps 650 and 1300. Nevertheless, the mean-field curve falls nicely within the standard deviation of the simulation results. Both curves approach the same stable state $\frac{c}{n} = 500$ items after 1300 steps.

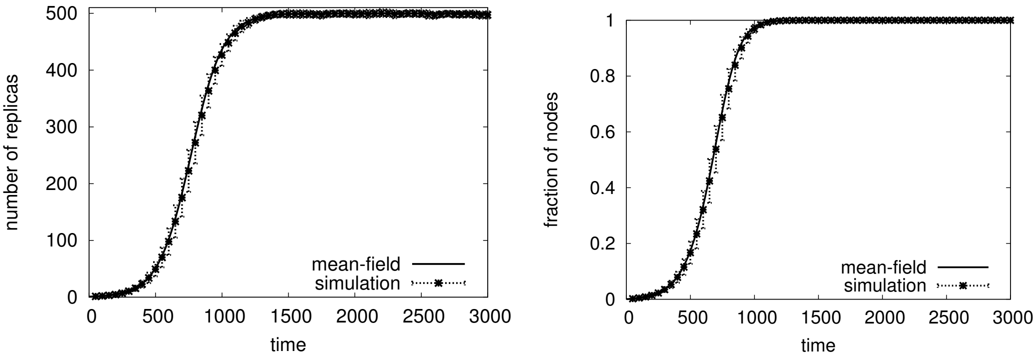
Figure 6(b) shows the evolution of the total number of nodes that have held the



(a) Replication

(b) Coverage

Fig. 6. Comparison with simulation results for a network of 2500 nodes.



(a) Replication

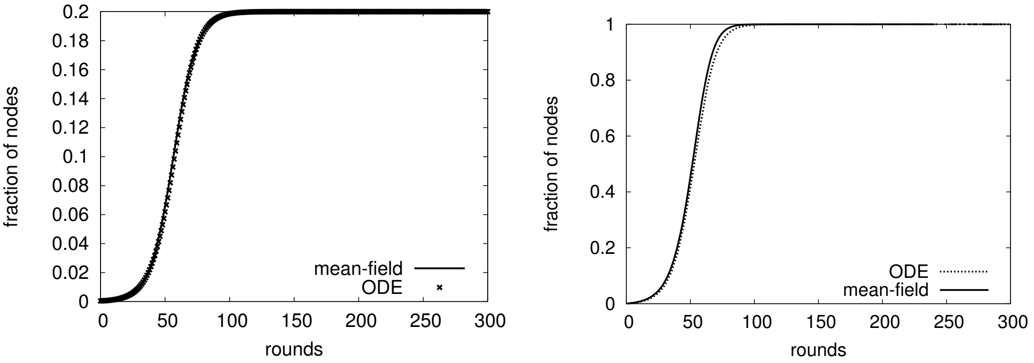
(b) Coverage

Fig. 7. Comparison with simulation results for a network of 25000 nodes.

distinguished item at some moment in time. In the mean-field model, we sum the fraction of nodes that are in state $(g, d, 1)$ for all g and d. Like replication, in the beginning both curves grow at the same rate for around 650 steps, after which there is a slight difference that accumulates over time. Still, the mean-field curve falls within the standard deviation of the simulation results, and both curves reach the stable state 1 by 1500 steps.

The difference between the simulation and mean-field curves in the network of 2500 nodes arises from the fact that the distribution of the items slightly fluctuates around the average 500 (average deviation is 20). As a consequence the uniform distribution assumed in our model is not perfect in practise. However, when considering larger networks (i.e. 25000 nodes) the simulations come very close to the behaviour predicted by the mean-field analysis; see Figure 7.

In [14], the state transition model is used for the differential equations for replication and coverage, presented in Sec. 2.3. Thus, we compare our mean-field model with the differential equations for both replication (see Figure 8(a)) and coverage (see Figure 8(b)). Due to the deterministic nature of both models, we are interested to see how different the results are. We implemented the differential equations in MATLAB; the solutions of the differential equations are obtained by numerical



(a) Replication

(b) Coverage

Fig. 8. Comparison of the deterministic models.

integration. The number of possible contacts by a node per round is limited to 4. To take into account the fact that the equational models disregard the possibility of collisions, we increase the maximal gossip delay for the mean-field analysis to $G_{\max} = 200$. Intuitively, there are $N/G_{\max}$ nodes that have the same gossip delay g, and nodes with $g = 0$ talk simultaneously. By increasing $G_{\max}$ and initially distributing all nodes uniformly at random into states (g, d, o) with different g, fewer nodes become active at the same time, which leads to more sequential initiation of a gossip by each node. For $G_{\max} = 200$ and $N = 2500$, $2500/200 = 12.5$ nodes have the same g compared to $2500/10 = 250$ nodes, thus having a smaller chance that an active node contacts another active node or a communicating pair than in the latter case. Formally, since the equational models do not include the notion of collisions, we have to maximize the probability of no collisions $\text{noc}(\mathbf{m})$ in the mean-field model. Maximising the value of the probability of no collision $e^{-2 \cdot \mathbf{m}_{(0,d,o)}}$ implies minimizing $\mathbf{m}_{(0,d,o)}$ since $e > 1$, which can be achieved by increasing $G_{\max}$. The maximal gossip delay $G_{\max} = 200$ keeps the probability of no collisions in the mean-field model fairly small, i.e. $1/e^{25}$, while preserving the relatively small size of the state space, i.e. $201 \times 2 \times 2 = 804$.

We assume that one round of the differential equation corresponds to 200 steps in the mean-field model, because in case of the equational models, every node has to initiate a gossip once per round.

Figure 8(a) shows the comparison of both models for replication, in terms of the evolution of the fraction of nodes that have seen the distinguished item over time. The differential equation curve overlaps nicely with the mean-field curve. Figure 8(b) presents the comparison of two models for the coverage property. Both curves are very close to each other, with a tiny difference starting after around 60 rounds; both curves settle to the same stable state after 80 rounds. The difference is explained by the assumed value for the number of maximum contacts by a node per round, and a small collision probability $1 - 1/e^{25}$ compared to 0 of the differential equation.

5 A Mean-field Model for Basic GTP

A detailed description of basic GTP can be found in Sec. 2.4 and in [29,28]. This corresponds to Step 1 in our methodology. Steps 2 and 3 are accomplished in the following three subsections (5.1–5.2). Sec. 5.3 corresponds to Steps 4 and 5. We compare the obtained mean-field model against the emulation results from [28] in Sec. 5.4. We perform an analysis of properties of basic GTP in Sec. 5.5, based on the mean-field model.

5.1 State Space

The state of a node in a basic GTP network is given by a triple (g, l, h). The first component g denotes the gossip delay, which defines the frequency of initiating a contact. When the value of g becomes equal to zero, a node initiates a gossip interaction. The second component l represents a counter for the last update. In GTP, the time of the last clock update is stored and, if necessary, compared to the current time. If the difference exceeds the standalone period, an update is enforced. We replace it with a counter which is set to the length of the standalone period at every update. Reaching zero, a clock update is enforced at the next interaction. Finally, h is the number of hops the timing information has travelled from the time source.

Let $G_{\max}$ be the maximal gossip delay, and let L be the standalone period. We introduce H to be the maximal hop count. A node in a state with $h = H$ has a hop count of *at least* H. A node with $h = \infty$ is said to be *unsynchronised*. The state space of single node then is

$$S = \{0, \ldots, G_{\max}\} \times \{0, \ldots, L\} \times \{0, \ldots, H, \infty\},$$

which is of size $|S| = (G_{\max} + 1)(L + 1)(H + 2)$.

Gossip delay

Though basic GTP has a fixed gossip delay, we design the model in such a way that it allows for the gossip delay to vary, depending on the hop count of a node. We assume that there is a function $G : \{0, \ldots, H, \infty\} \mapsto \{0, \ldots, G_{\max}\}$ that gives the gossip delay $G(h)$ for any hop count h. As described in Sec. 2.4, the varying gossip delay is an optimization feature of gradual GTP.

5.2 Local Transition Probabilities

The behaviour of a single node is determined by its state and the current occupancy measure $\mathbf{m}$. In the sequel, we again use a kind of pattern matching notation: for example, $(g, l, h \mid g > 0)$ denotes any state where $g > 0$ while l and h are chosen arbitrarily from their respective value sets. The expression $\mathbf{m}_{(g,l,h|h<H)}$ is an example for the abbreviation of a sum of occupancy fractions, defined by

$$\mathbf{m}_{(g,l,h|h<H)} = \sum_{g} \sum_{l} \sum_{h<H} \mathbf{m}_{(g,l,h)}.$$

Time sources

We begin with the description of the behaviour of a time source, that is, a node in a state with $h = 0$. Time sources never update their clock, hence, component l has no meaning and we always set it to L. If the gossip delay is larger than zero ($g > 0$), we just decrement it by one. If it is equal to zero, the gossip delay is reset to $G(0)$.

$$\begin{aligned} P^N_{(g,l,0|g>0),(g-1,L,0)}(\mathbf{m}) &= 1 \\ P^N_{(0,l,0),(G(0),L,0)}(\mathbf{m}) &= 1. \end{aligned}$$

As one can see, time sources act independently of their environment (as depicted in Fig. 9).

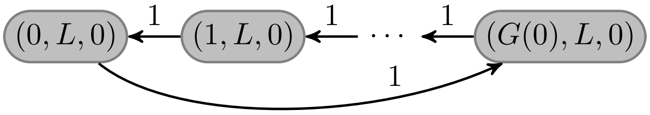


Fig. 9. Transitions of time sources.

Active nodes

If the gossip delay g of a node A is equal to zero, it becomes *active* and initiates a gossip interaction with a peer B randomly chosen from the remaining $N - 1$ nodes. In this interaction, the clock of A might get updated. In GTP, an interaction is discarded if during its course there has been another interaction leading to an update of the clock. In the model we require that for each node only one interaction can be active, otherwise we say that there is a collision. An update can only take place if the interaction prevails, that is, no collision occurs. After A has chosen a suitable peer B, the probability $\mathrm{noc}^N(\mathbf{m})$ that there is **no** collision is given by the probability that all other active nodes select peers different from A and B. The probability that a node chooses neither A nor B (given that it does not try to interact with itself) is $(N - 3)/(N - 1)$. We consequently have

$$\mathrm{noc}^N(\mathbf{m}) = \left(\frac{N-3}{N-1}\right)^{\mathbf{m}_{(0,l,h)} \cdot N - 1} .$$

We further have to distinguish between nodes with an enforced update ($l = 0$) and those without. If an update is enforced, the clock will be updated as long as the peer is synchronised, having a hop count h' different from ∞. If the update is optional, the clock value is only changed if this does not increase the hop count, that is, if $h' < h$. In either case, the new state after a successful update is $(G(h'+1), L, h'+1)$. The probability to select a passive peer with hop count h' is $\mathbf{m}_{(g',l',h'|g>0)} \cdot N/(N-1)$ and so the probability of a successful update is

$$P^N_{(0,0,h|h>0),(G(h'),L,h'+1)}(\mathbf{m}) = \frac{\mathbf{m}_{(g',l',h'|g'>0)} \cdot N}{N-1} \cdot \mathrm{noc}^N(\mathbf{m}), \quad \forall h' < \infty,$$

$$P^N_{(0,l,h|l>0,h>0),(G(h'),L,h'+1)}(\mathbf{m}) = \frac{\mathbf{m}_{(g',l',h'|g'>0)} \cdot N}{N-1} \cdot \mathrm{noc}^N(\mathbf{m}), \quad \forall h' < h.$$

Clearly, the successful update takes place if no collision occurred during the interaction (i.e. with the probability $\mathrm{noc}^N(\mathbf{m})$). If $h = \infty$ and the active node chooses a passive peer with hop count $H - 1$ or H, the probability of the successful update is:

$$P^N_{(0,l,\infty),(G(H),L,H)}(\mathbf{m}) = \sum_{i=0}^{1} \frac{\mathbf{m}_{(g',l',H-i\ |g'>0)} \cdot N}{N-1} \cdot \mathrm{noc}^N(\mathbf{m}),$$

since the interaction occurred with a synchronised peer.

The case remains where the interaction does not lead to an update of the clock. This can happen if (1) a collision occurs when gossiping with a passive node, (2) an active node is selected as peer, also leading to a collision, or (3) the interaction has prevailed but the peer cannot provide a suitable hop count; that is, the peer is an unsynchronised node if $l = 0$, and $h' \geq h$ if $l > 0$. We again distinguish enforced and optional updates and get the following probabilities:

$$\begin{aligned}
P^N_{(0,0,h|h>0),(G(h),0,h)}(\mathbf{m}) &= \frac{\mathbf{m}_{(g',l',h'|g'>0)} \cdot N}{N-1} \cdot (1 - \mathrm{noc}^N(\mathbf{m})) \\
&\quad + \frac{\mathbf{m}_{(0,l',h')} \cdot N}{N-1} + \frac{\mathbf{m}_{(g',l',\infty|g'>0)} \cdot N}{N-1} \cdot \mathrm{noc}^N(\mathbf{m}), \\
P^N_{(0,l,h|l>0,h>0),(G(h),l-1,h)}(\mathbf{m}) &= \frac{\mathbf{m}_{(g',l',h'|g'>0)} \cdot N}{N-1} \cdot (1 - \mathrm{noc}^N(\mathbf{m})) \\
&\quad + \frac{\mathbf{m}_{(0,l',h')} \cdot N}{N-1} + \frac{\mathbf{m}_{(g',l',h'|g'>0,\mathbf{h'}\geq\mathbf{h})} \cdot N}{N-1} \cdot \mathrm{noc}^N(\mathbf{m}).
\end{aligned}$$

We treat the case where an active node with hop count H contacts another node with hop count H as an unsuccessful update, since in our model H subsumes all hop counts $h \geq H$ of the synchronised nodes of the actual protocol. The calculation of the probability of the successful update in this case requires knowledge of the actual distribution of nodes with these hop counts, which we do not know apriori.

Passive nodes

A passive node with $g > 0$ has to be contacted by an active peer with hop count h' to be able to update its hop count to $h' + 1$. This happens with probability $\mathbf{m}_{(0,l,h')} \cdot N/(N-1)$. The gossip delay is decreased by one in all cases, shortening the time until the next gossip initiation. Following the same line of argumentation as for active nodes, the probabilities for successful interactions are

$$P^N_{(g,0,h|g>0,h>0),(G(h'+1),L,h'+1)}(\mathbf{m}) = \frac{\mathbf{m}_{(0,l',h')} \cdot N}{N-1} \cdot \mathrm{noc}^N(\mathbf{m}), \quad \forall h' < \infty,$$

$$P^N_{(g,l,h|g>0,l>0,h>0),(G(h'+1),L,h'+1)}(\mathbf{m}) = \frac{\mathbf{m}_{(0,l',h')} \cdot N}{N-1} \cdot \text{noc}^N(\mathbf{m}), \quad \forall h' < h.$$

That is, the probability of being selected by an active peer with hop count h' is $\mathbf{m}_{(0,l',h')} \cdot N/(N-1)$, and the probability that no collision occurred during the interaction is $\text{noc}^N(\mathbf{m})$.

If $h = \infty$ and the passive node is chosen by an active peer with hop count $H-1$ or H, the probability of the successful update is:

$$P^N_{(g,l,\infty|g>0),(G(H),L,H)}(\mathbf{m}) = \sum_{i=0}^{1} \frac{\mathbf{m}_{(0,l',H-i)} \cdot N}{N-1} \cdot \text{noc}^N(\mathbf{m}),$$

because the passive node is contacted by a synchronised peer.

The probability of not updating the clock is again composed of three terms: (1) the probability of having a collision during the interaction with an active node, (2) the probability of not being chosen as a peer by any active node, and (3) the probability of having an interaction with an active peer that does not have a suitable hop count: either unsychnronised node if $l = 0$, or with hop count $h' \geq h$ if $l > 0$. Hence,

$$\begin{aligned}
P^N_{(g,0,h|g>0,h>0),(g-1,0,h)}(\mathbf{m}) &= \frac{\mathbf{m}_{(0,l',h')} \cdot N}{N-1} \cdot (1 - \text{noc}^N(\mathbf{m})) \\
&\quad + \left(1 - \frac{\mathbf{m}_{(0,l',h')} \cdot N}{N-1}\right) + \frac{\mathbf{m}_{(0,l',\infty)} \cdot N}{N-1} \cdot \text{noc}^N(\mathbf{m}), \\
P^N_{(g,l,h|g>0,l>0,h>0),(g-1,l-1,h)}(\mathbf{m}) &= \frac{\mathbf{m}_{(0,l',h')} \cdot N}{N-1} \cdot (1 - \text{noc}^N(\mathbf{m})) \\
&\quad + \left(1 - \frac{\mathbf{m}_{(0,l',h')} \cdot N}{N-1}\right) + \frac{\mathbf{m}_{(0,l',h'|\mathbf{h'\geq h})} \cdot N}{N-1} \cdot \text{noc}^N(\mathbf{m}).
\end{aligned}$$

Again, we treat the case where a passive node with hop count H is chosen by an active peer with hop count H as an unsuccessful update, since in our model H subsumes all hop counts $h \geq H$ of the synchronised nodes of the actual protocol. Like in the case of active nodes, the calculation of the probability of the successful update in this special case requires knowledge of the actual distribution of nodes with these hop counts, which we do not know apriori.

5.3 Mean-field Limits

The probability $\text{noc}^N(\mathbf{m})$ of having no collision converges for $N \to \infty$:

$$\text{noc}(\mathbf{m}) = \lim_{N\to\infty} \text{noc}^N(\mathbf{m}) = \lim_{N\to\infty} \left(\frac{N-3}{N-1}\right)^{\mathbf{m}_{(0,l,h)} \cdot N - 1} = e^{-2 \cdot \mathbf{m}_{(0,l,h)}}.$$

For all the local transition probabilities the number of nodes N only appears in the factor $N/(N-1)$, which has limit 1 for $N \to \infty$. The limit probabilities are thus easily obtained by removing the factor $N/(N-1)$ from the expressions and by replacing $\mathrm{noc}^N(\mathbf{m})$ by the above limit $\mathrm{noc}(\mathbf{m})$.

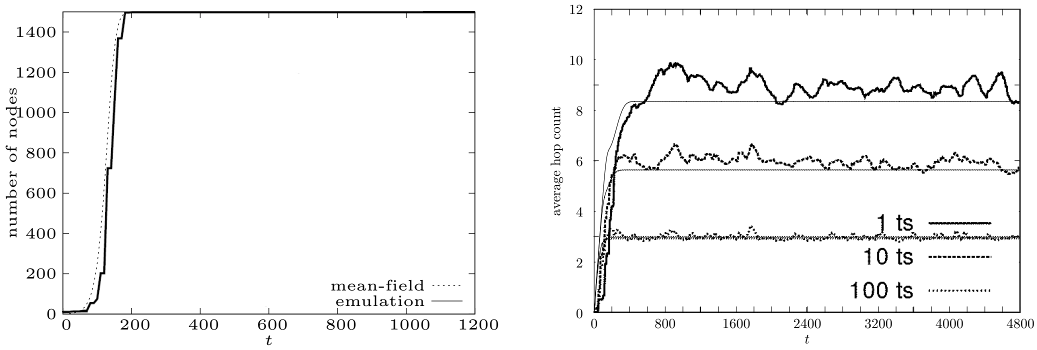
5.4 Comparison with Emulation Results

In [28], emulation is used to explore how GTP behaves in practice. For basic GTP a network of 1500 nodes was emulated on a single workstation, using local object passing implementation for communication. One node is a time source, having hop count zero, all other nodes are not synchronised. The gossip delay is fixed and independent of the state of a node and set to 25 seconds. The maximum standalone period is also set to 25 seconds.

Fitting our model to this scenario, we set the fraction of nodes being a time source to $1/1500$. We assume that one step in the model corresponds to one second in the emulation. This slightly overestimates the duration of a gossip interaction, which is reported to be in the sub-second range. The maximum gossip delay is $G_{\max} = 25$ seconds, and since it is fixed we have $G(h) = G_{\max}$ for any hop count h. The maximum delay between two updates is $L = 25$ seconds. The maximum hop count is chosen to be $H = 15$. A single node thus can assume $26 \cdot 26 \cdot 17 = 11492$ states. The time source fraction starts off with $g = 12$, that is, it initiates a gossip interaction for the first time after 12 seconds. The unsynchronised nodes have remaining gossip delays uniformly distributed between 0 and $G_{\max}$.

Figure 10(a) shows the evolution of the number of nodes that are *aware* of the time source over time. A node becomes *aware* of the time source existence when its hop count changes to a finite value. For the mean-field model we have multiplied the fraction of nodes with a hop count smaller than ∞ by 1500 to obtain the depicted curve. The curves of emulation and analytical model proceed close to each other, both approaching 1500 after about 200 seconds, that is, after about 8 gossip cycles.

In Figure 10(b) we compare the evolution of the average hop count when the number of time sources is multiplied by 10 and 100, respectively. For the mean-field



(a) Discovering time source existence

(b) Avg. hop count for different number of time sources

Fig. 10. Comparison with emulation results (data taken from Figs. 6.1(a), 6.5(b) in [28])

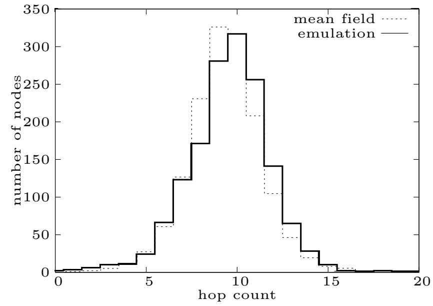


Fig. 11. Distribution of hop count (emulation results from Figure 6.6 in [28])

model, the average hop count is computed for synchronised nodes only, thus being a kind of underestimation as long as not all nodes are aware of the time source existence. At the beginning, the average hop count increases faster in the mean-field, however, mean-field model and emulation settle to similar values. The change in the average hop count depends logarithmically on the number of time sources. Moreover, having the higher number of time sources result in the nonzero elements in the occupancy measure $\mathbf{m}$, have values closer to each other.

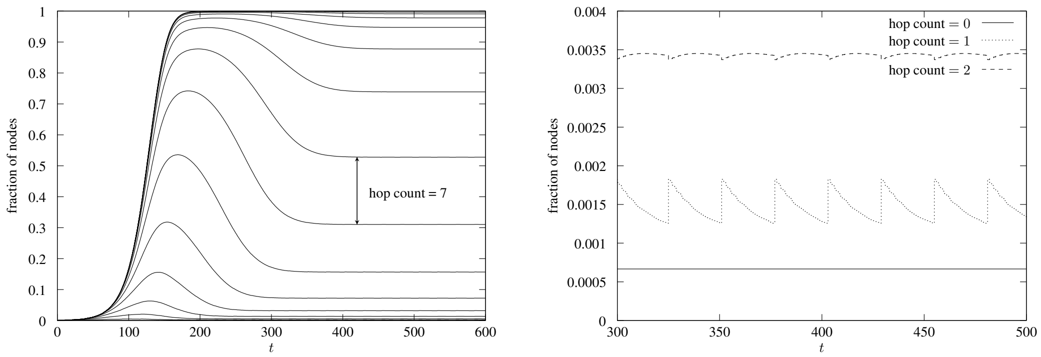
Figure 11 finally shows the histogram of hop counts after the protocol stabilises. For the mean-field curve we have taken the distribution at time $t = 600$, neglecting the fact that there are minor oscillations because of time source gossip. Taking this into account, and the fact that we are considering a fully connected network in the mean-field model in contrast with a special peer sampling service in the emulations, there is a close match between the emulation and the mean-field result.

Figures 10 and 11 document that the presented mean-field model captures the main features of basic GTP. The evolution of the hop counts in the considered scenario is quite precisely represented. In the following we concentrate on the mean-field model. We show further measures that were not considered in the emulation experiments in [28] and also evaluate the influence of varying the gossip delay.

5.5 More Properties of Basic GTP

We stick to the scenario of the previous section. Following Figure 11 we depict in Figure 12(a) the evolution of the hop count distribution over the first 10 minutes. Each curve corresponds to the fraction of nodes that have a hop count of *at most* a given value. The distance between two curves corresponds to the fraction of nodes that have exactly a given hop count, as shown for a hop count of seven.

At the beginning, almost all nodes are unsynchronised, which results in the area to the left of the graph. Gradually, the nodes acquire hop counts smaller than ∞. Since this change originates from the time source, hop counts different from ∞ are relatively small at first. Over time, the hop count distribution settles to a quasi stable state, with – on average – higher hop counts than at the moment the network got fully synchronised.



(a) Distribution of hop count

(b) Periodicity of hop count probabilities

Fig. 12. Hop counts for constant gossip delay (25 seconds)

Figure 12(a) suggests that the hop count distribution reaches a stable state. This is not completely true, since there is a periodic disturbance whenever the time source fraction gossips every 25 seconds. Figure 12(b) shows the fraction of nodes having hop count zero, one, or two for the time interval from 300 to 500 seconds. The fraction of time sources (hop count 0) is constant since no node with a hop count larger than zero can ever become a time source. The fraction of nodes having hop count one oscillates: with each time source gossip, the fraction increases abruptly, decreasing gradually afterwards. Also for hop count two, there is still a visible periodicity. The change is already very small, for higher hop counts the periodicity effect wears off completely.

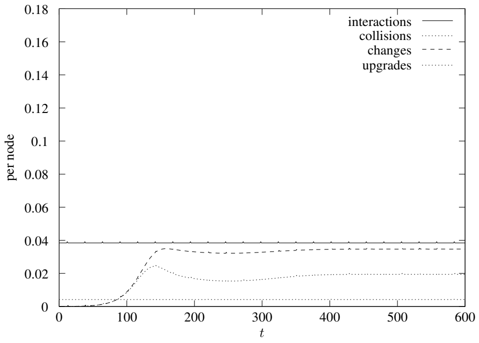


Fig. 13. Interaction activity with gossip delay $G_{\min} = G_{\max} = 25$

Figure 13 depicts the interaction activity per node over the first 10 minutes. The nature of the mean-field model makes it necessary to state interactions *per node*. Mapping back to the original scenario, multiplying the indicated values with 1500 results in the total number per second. *Interactions* are gossip attempts by nodes which have reached hop count zero. The number stays constant due to the fact that we have a constant gossip delay of 25 seconds. Only when the time source fraction gossips, there is a small spike in the curve. A *collision* occurs if there is more than one attempt of a gossip interaction with a node. The number of collisions is also constant, being a function of the number of interactions. A *change* means that a node adjusts its clock and hop count to a different value. The hop count might be larger than before if the update has been been forced by the last-update flag. At the beginning, changes are rare, since most interactions take place between unsynchronised nodes, not leading to any updates. In the synchronisation phase, the number of changes increases and finally settles down to a stable level. Because of enforced updates there are always changes to be expected. With *upgrades* we denote changes that actually lead to a better hop count. Their number is of course smaller than the number of changes, settling down to a positive number as well: as long as there are 'downgrading' changes because of enforced updates, there will also be upgrading interactions in the sequel.

Later in the paper, we compare this graph to the mean-field results with the other values of the gossip delays $G_{\min}$ and $G_{\max}$ in order to show the impact of different values on the protocol performance. Thus, the scale of the y-axis is chosen to be the same for the convenient comparison of these results.

Finally we show the behaviour of the network when the time source fails. We do this by starting with a single time source, and removing it after 10 minutes (time $t = 600$). That is, by redistributing the fraction of nodes being a time source (i.e. with hop count 0) to the other states with hop counts > 0. Figure 14 shows what happens to the hop count distribution in the following 10 minutes. In this figure, like in Figure 10(a), each curve represents the fraction of nodes that have at most a certain hop count. Thus, the highest curve on the graph corresponds to the fraction of nodes with hop count at most $H = 15$, i.e., the fraction of synchronised nodes. As could be expected, nodes with a low hop count die out over time, leaving all nodes at the chosen maximum hop count of $H = 15$.

5.6 *Different Static Gossip Delays*

With the next graphs we want to clarify the influence of the gossip delay on the performance of the complete network. In general, one expects that a higher gossip delay leads to a lower synchronisation speed. On the other hand, it also implies less communication. The question we asked ourselves was: can a short gossip delay lead to a slower synchronisation than a longer gossip delay because of too many

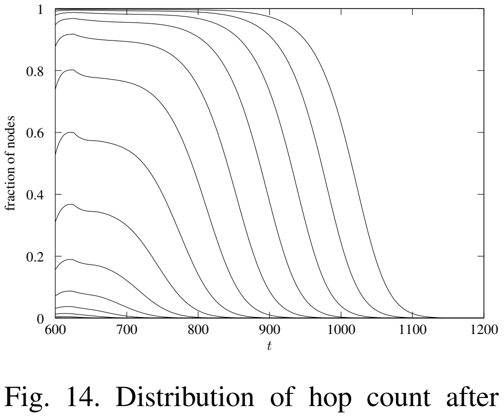


Fig. 14. Distribution of hop count after time source fails for constant gossip delay (25 seconds).

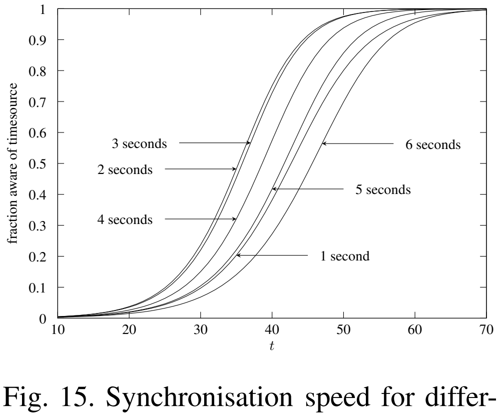
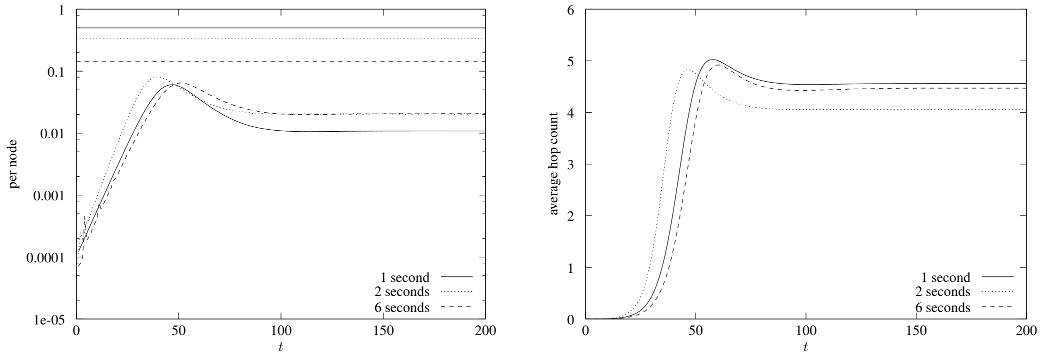


Fig. 15. Synchronisation speed for different static gossip delays



(a) Upgrading interactions

(b) Average hop count

Fig. 16. Varying the static gossip delay

interactions and, subsequently, collisions?

Figure 15 shows the speed of synchronisation for gossip delays between one and six seconds. In general, synchronisation slows down with increasing gossip delay. But if nodes gossip every other second, that is, if the gossip delay is one, synchronisation proceeds slower than for a delay of two, three, four or five seconds. In this case, collisions impede a fast dissemination of the timing information through the network.

Figures 16(a) and 16(b) further substantiate this insight. In 16(a), the upper set of curves depicts the total number of initiated interactions, and the lower set shows the number of interactions really leading to an upgrade. Even though with a gossip delay of one second the total number of interactions is highest, the number of upgrades is lower than for a gossip delay of two seconds. Figure 16(b) documents that also the average hop count for a gossip delay of one is higher than for a gossip delay of two.

5.7 Dynamic Adaptation of Gossip Delay

Our model allows us to dynamically adapt the gossip delay to the state of the system, that is, to its current hop count, via the function $G(h)$. *Gradual* GTP also offers this possibility, thereby taking more parameters (not only the hop count) into account. In line with the description in [28] we want a node to gossip more often if its time has “bad quality”. For our model this translates to a high hop count.

For this purpose we introduce a minimal gossip delay $G_{\min}$. The gossip delay of a node is then set to

$$G(h) = \begin{cases} G_{\min}, & h = \infty, \\ G_{\max} - \lfloor \frac{h}{H+1} \rfloor \cdot (G_{\max} - G_{\min}), & 0 \le h \le H. \end{cases}$$

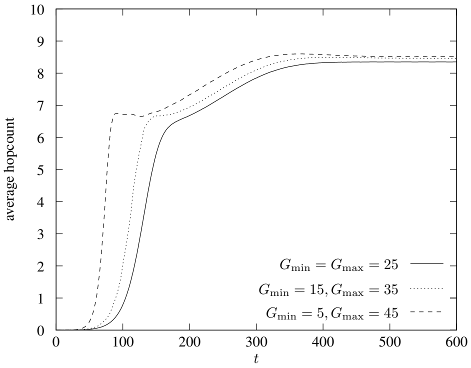


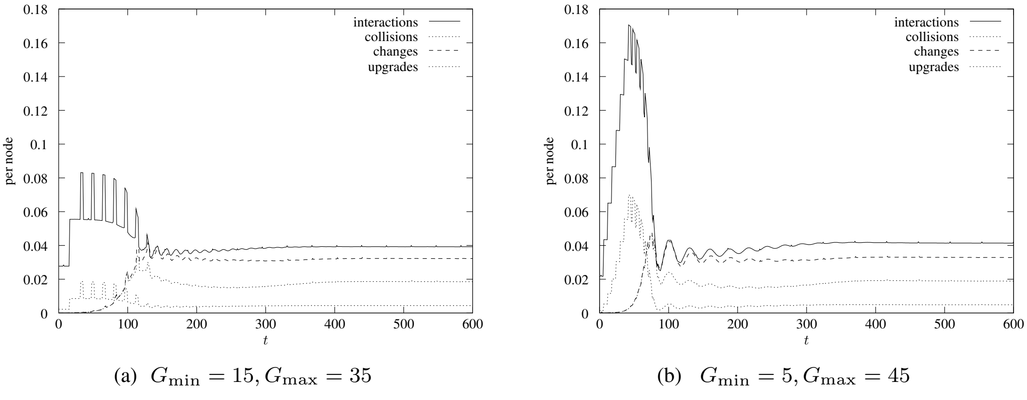
Fig. 17. Synchronisation speed and average hop count.

Unsynchronised nodes initiate a gossip as often as possible while a time source does so as seldom as possible. For all other hop counts, the gossip delay spreads linearly between $G_{\min}$ and $G_{\max}$. Note that numerically a hop count of ∞ is treated like $H + 1$.

We compare three cases: the scenario considered so far with $G_{\min} = G_{\max} = 25$ seconds (see Figure 13), a small range of gossip delays with $G_{\min} = 15$ seconds and $G_{\max} = 35$ seconds, and a large range with $G_{\min} = 5$ seconds and $G_{\max} = 45$ seconds.

Figure 17 depicts both the synchronisation speed (left set of curves) and the average hop count (right set of curves) in the first 10 minutes. Since nodes with a high hop count have more chances to upgrade if the range of the gossip delay is enlarged, the synchronisation speed is increased as opposed to the static gossip version. But this higher gossiping frequency of nodes with a high hop count also leads to a slightly increased average hop count.

Figures 18(a) and 18(b) compare the interaction activity of two scenarios. While for a static gossip delay the number of interactions is independent of the state of the system (as shown in Figure 13), it highly depends on the state if the gossip delay is computed dynamically. In the beginning, when most of the nodes are unsynchronised, there are many interactions, leading to faster synchronisation. In the



(a) $G_{\min} = 15, G_{\max} = 35$

(b) $G_{\min} = 5, G_{\max} = 45$

Fig. 18. Interaction activity.

long run, the activity pattern settles to similar values for all three scenarios, with a slightly increasing number of interactions with increasing range of gossip delay.

6 Conclusion

The main motivation for developing a modelling methodology for gossip protocols is that, although these protocols are appealing with respect to scalability, robustness, and simplicity, it is hard to quantitatively predict the performance according to a particular metric or analyse further possible optimisations and limitations analytically.

We have demonstrated that mean-field analysis is suitable for gossip protocols. The following premises enable mean-field analysis:

- there is a very large number of identically behaving nodes (symmetry property [9]);
- there are no central servers or global resources;
- the behaviour of a single node can be described in a local way;
- the number of states of a single node is small in comparison to the number of nodes;
- transient measures ("at time t") are of interest.

Extensions of the theory presented here would also allow for the incorporation of a global memory, the failure or entering/leaving of nodes [20], the employment of continuous-time models, and steady-state measures [17]. However, the mean-field approach does not allow for the evaluation of a centrally managed network or the separate modelling of one single node.

In this paper we considered two applications of gossip protocols: along with the presentation of the necessary theory, we developed a simple information dissemination model. The suitability of the mean-field approximation method was shown by comparing the results obtained by analytically solving the resulting DTMC, by computing the mean-field limit, and by simulating the system. The mean-field method is applied to the bidirectional (push-pull) communication model for the case of a more complex application of gossip-based information dissemination, shuffling protocol.

As a larger case study for an aggregating gossip protocol we derived a mean-field model for basic GTP. It includes the hop count metric and the constant gossip delay, and also takes into account enforced updates due to the expiration of the standalone period. We validated the fit of the mean-field model matching it to emulation results taken from [28]. Then we used the mean-field model to derive a variety of interesting measures, also considering dynamically adjusted gossip delays.

With the conducted experiments we have shown how to successfully derive a large variety of useful measures for basic GTP using a mean-field model. While emulation like in [28] requires the availability of suitable hardware and runs in real-time (20 minutes for most shown measures), the evaluation of the mean-field model for a given parameter setting is done in a couple of minutes.

For future work, we plan to investigate mean-field analysis for alternative stochastic models for the nodes, e.g., by moving to the continuous-time context [17] or by introducing non-determinism using Markov decision processes [44].

References

[1] L. Afanassieva, S. Popov, G. Fayolle, Models for transporation networks., J. of Math. Sciences 84 (3) (1997) 1092–1103.

[2] A. Allavena, A. Demers, J. Hopcroft, Correctness of a gossip based membership protocol, in: Proc. ACM Symp. on Principles of Distributed Computing, ACM Press, 2005, pp. 292–301.

[3] H. Andersson, T. Britton, Stochastic Epidemic Models and Their Statistical Analysis, vol. 151 of Lecture Notes in Statistics, Springer-Verlag, 2000.

[4] J. P. Aparicio, M. A. Natiello, H. G. Solari, The quasi-deterministic limit of population dynamics., presented at Int. ISAAC Congress, July 25-30, 2005. Preprint (2007).

[5] F. Baccelli, A. Chaintreau, D. De Vleeschauwer, D. McDonald, A mean-field analysis of short lived interacting TCP flows, SIGMETRICS Perform. Eval. Rev. 32 (1) (2004) 343–354.

[6] F. Baccelli, A. Chaintreau, D. De Vleeschauwer, D. McDonald, HTTP turbulence., AMS Networks and Heterogeneous Media 1 (2006) 1–40.

[7] F. Baccelli, D. McDonald, M. Lelarge, Metastable regimes for multiplexed TCP flows., in: Allerton Conf. on Communication, Control, and Computing, 2004, pp. 1005–1011.

[8] F. Baccelli, D. McDonald, J. Reynier, A mean-field model for multiple TCP connections through a buffer implementing RED, Perform. Eval. 49 (1-4) (2002) 77–97.

[9] R. Bakhshi, F. Bonnet, W. Fokkink, B. Haverkort, Formal analysis techniques for gossiping protocols, SIGOPS Oper. Syst. Rev. 41 (5) (2007) 28–36.

[10] R. Bakhshi, L. Cloth, W. Fokkink, B. R. Haverkort, Mean-field analysis for the evaluation of gossip protocols, SIGMETRICS Perform. Eval. Rev. 36 (3) (2008) 31–39.

[11] R. Bakhshi, L. Cloth, W. Fokkink, B. R. Haverkort, Mean-field analysis for the evaluation of gossip protocols, in: Proc. Conf. on Quantitative Evaluation of SysTems (QEST), IEEE Computer Society, 2009, pp. 247–256.

[12] R. Bakhshi, J. Endrullis, S. Endrullis, W. Fokkink, B. Haverkort, Automating the mean-field method for large dynamic networks, in: Proc. Conf. on Quantitative Evaluation of SysTems (QEST), IEEE Computer Society, 2010, to appear.

[13] R. Bakhshi, A. Fehnker, On the impact of modelling choices for distributed information spread. a comparative study, in: Proc. Conf. on Quantitative Evaluation of SysTems (QEST), IEEE Computer Society, 2009, pp. 41–50.

[14] R. Bakhshi, D. Gavidia, W. Fokkink, M. van Steen, An analytical model of information dissemination for a gossip-based protocol, Computer Networks. Special Issue on Gossiping in Distributed Systems 53 (13) (2009) 2288–2303.

[15] R. J. Baxter, Exactly Solved Models in Statistical Mechanics, Academic Press, 1982.

[16] M. Benaim, J.-Y. Le Boudec, A Class Of Mean Field Interaction Models for Computer and Communication Systems, Performance Evaluation 65 (11-12) (2008) 823–838.

[17] A. Bobbio, M. Gribaudo, M. Telek, Analysis of large scale interacting systems by mean field method, in: Proc. Conf. on Quantitative Evaluation of Systems, IEEE, 2008, pp. 215–224.

[18] F. Bonnet, Performance analysis of Cyclon, an inexpensive membership management for unstructured P2P overlays., Master's thesis, ENS Cachan Bretagne, University of Rennes, IRISA (2006).

[19] E. Bortnikov, M. Gurevich, I. Keidar, G. Kliot, A. Shraer, Brahms: Byzantine resilient random membership sampling, Computer Networks. Special Issue on Gossiping in Distributed Systems 53 (13) (2009) 2340–2359.

[20] J.-Y. L. Boudec, D. McDonald, J. Mundinger, A generic mean field convergence result for systems of interacting objects, in: Proc. Conf. on the Quantitative Evaluation of Systems, IEEE, 2007, pp. 3–18.

[21] A. Chaintreau, J.-Y. Le Boudec, N. Ristanovic, The age of gossip: spatial mean field regime, in: Proc. Conf. on Measurement and Modeling of Computer Systems (SIGMETRICS), ACM, 2009, pp. 109–120.

[22] P. Costa, V. Gramoli, M. Jelasity, G. P. Jesi, E. Le Merrer, A. Montresor, L. Querzoni, Exploring the interdisciplinary connections of gossip-based systems, SIGOPS Oper. Syst. Rev. 41 (5) (2007) 51–60.

[23] D. Dawson, J. Tang, Y. Zhao, Balancing queues by mean field interaction, Queueing Syst. Theory & Appl. 49 (3–4) (2005) 335–361.

[24] S. Deb, M. Médard, C. Choute, Algebraic gossip: a network coding approach to optimal multiple rumor mongering, IEEE/ACM Trans. Netw. 14 (SI) (2006) 2486–2507.

[25] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, D. Terry, Epidemic algorithms for replicated database maintenance, in: Proc. ACM Symp. on Principles of Distributed Computing, ACM Press, 1987, pp. 1–12.

[26] S. N. Ethier, T. G. Kurtz, Markov Processes: Characterization and Convergence, Wiley, 1986.

[27] D. Gavidia, S. Voulgaris, M. van Steen, A Gossip-based Distributed News Service for Wireless Mesh Networks., in: Proc. Conf. on Wireless On Demand Network Syst. and Services, IEEE, 2006, pp. 59–67.

[28] K. Iwanicki, Gossip-based dissemination of time, Master's thesis, Warsaw University and Vrije Universiteit Amsterdam (2005).

[29] K. Iwanicki, M. van Steen, S. Voulgaris, Gossip-based clock synchronization for large decentralized systems, in: Proc. Workshop on Self-Managed Networks, Systems and Services, vol. 3996 of LNCS, Springer, 2006, pp. 28–42.

[30] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, M. van Steen, The peer sampling service: Experimental evaluation of unstructured gossip-based implementations, in: Proc. ACM/IFIP/USENIX Middleware Conf., vol. 3231 of LNCS, Springer, 2004, pp. 79–98.

[31] M. Jelasity, W. Kowalczyk, M. van Steen, Newscast computing., Tech. Rep. IR-CS-006, Vrije Universiteit Amsterdam (2003).

[32] M. Jelasity, A. Montresor, O. Babaoglu, Gossip-based aggregation in large dynamic networks, ACM Trans. Comput. Syst. 23 (3) (2005) 219–252.

[33] M. Jelasity, A. Montresor, G. P. Jesi, S. Voulgaris, PeerSim: A peer-to-peer simulator., `http://peersim.sourceforge.net/`.

[34] W. Kang, F. Kelly, N. Lee, R. Williams, Fluid and Brownian approximations for an internet congestion control model, in: Proc. Conf. on Decision and Control, vol. 4, 2004, pp. 3938–3943.

[35] F. Karpelevich, E. Pechersky, Y. Suhov, Dobrushin's approach to queueing network theory., J. of Applied Mathematics and Stochastic Analysis 9 (4) (1996) 373–397.

[36] A.-M. Kermarrec, M. van Steen, Gossiping in distributed systems, SIGOPS Oper. Syst. Rev. 41 (5) (2007) 2–7.

[37] S. Kumar, L. Massoulié, Integrating streaming and file-transfer internet traffic: fluid and diffusion approximations, Queueing Syst. Theory Appl. 55 (4) (2007) 195–205.

[38] T. G. Kurtz, The relationship between stochastic and deterministic models for chemical reactions, J. Chem. Phys. 57 (1972) 2976–2978.

[39] M. Kwiatkowska, G. Norman, D. Parker, Analysis of a gossip protocol in PRISM, SIGMETRICS Perform. Eval. Rev. 36 (4) (2008) 17–22.

[40] A. Martinoli, K. Easton, W. Agassounon, Modeling Swarm Robotic Systems: A Case Study in Collaborative Distributed Manipulation, Int. Journal of Robotics Research 23 (4) (2004) 415–436.

[41] T. Meyer, C. Tschudin, Chemical networking protocols, in: Proc. ACM Workshop on Hot Topics in Networks (HotNets-VIII), 2009.

[42] D. Mollison (ed.), Epidemic Models: Their Structure and Relation to Data, Cambridge University Press, 1995.

[43] M. Opper, D. Saad (eds.), Advanced Mean Field Methods: Theory and Practice, MIT Press, 2001.

[44] M. Puterman, Markov Decision Processes., Wiley, 1994.

[45] H. Solari, M. Natiello, Poisson approximation to density dependent stochastic processes: A numerical implementation and test, in: Proc. Workshop on Dynamical Systems from Number Theory to Probability-II, vol. 6 of Math. Modelling, Växjö Univ. Press, 2003, pp. 79–94.

[46] I. Stojanovic, M. Sharif, D. Starobinski, Data dissemination in wireless broadcast channels: Network coding or cooperation, in: Proc. Conf. on Information Sciences and Systems, IEEE, 2007, pp. 265–270.

[47] P. Tinnakornsrisuphap, A. Makowski, Limit behavior of ECN/RED gateways under a large number of TCP flows, in: Proc. of IEEE INFOCOM, vol. 2, IEEE, 2003, pp. 873–883.

[48] S. Voulgaris, D. Gavidia, M. van Steen, Cyclon: Inexpensive membership management for unstructured P2P overlays., J. Network and Syst. Manage. 13 (2) (2005) 197–217.

[49] N. Vvedenskaya, Y. Suhov, Dobrushin's mean-field approximation for a queue with dynamic routing., Markov Proc. Rel. Fields 3 (4) (1997) 493–526.

[50] Q. Zhang, D. Agrawal, Dynamic probabilistic broadcasting in MANETs., J. of Parallel and Distributed Computing 65 (2) (2005) 220–233.

[51] X. Zhang, G. Neglia, J. Kurose, D. Towsley, Performance modeling of epidemic routing, Comput. Netw. 51 (10) (2007) 2867–2891.