

# Mean-Field Analysis for the Evaluation of Gossip Protocols

Rena Bakhshi\*, Lucia Cloth<sup>†</sup>, Wan Fokkink\*, Boudewijn Haverkort<sup>†‡</sup>

\*Department of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, Netherlands

<sup>†</sup>Centre for Telematics & Information Technology, University of Twente, Enschede, Netherlands

<sup>‡</sup>Embedded Systems Institute, Eindhoven, 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, that 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 models. 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.

## I. INTRODUCTION

We consider large-scale networks where a large number of nodes interacts. In such networks, gossip protocols have shown to be a sensible paradigm for developing scalable and reliable communication mechanisms. For instance, information can be spread in a large-scale network if nodes periodically contact each other in a random fashion, and exchange their local information. When a large number of nodes interacts in a connected environment, various phenomena emerge that cannot be explained in terms of the behaviour of a single node. Therefore, we are interested in going from a detailed local model at node level to an abstract global model of the system.

To study the emergent behaviour of gossip protocols demands the consideration of large-scale networks [1]. Thus, the analysis of gossip protocols with automated tools is hard – it is, for example, beyond the capabilities of current probabilistic model-checking tools [2]. In this paper, we show that mean-field analysis is well suited for a formal evaluation of gossip protocols. The stochastic process representing the modelled system converges to a deterministic process if the number of nodes goes to infinity, providing an approximation for large numbers of nodes.

A preliminary version of this paper appeared in [3]. Notably, Sec. IV of the current paper presents an analysis of basic GTP, whereas [3] restricted the analysis to only the hop-count metric within basic GTP.

This paper is further organised as follows. Sec. II gives a brief overview of the gossip paradigm, and explains an instance of such a protocol, that is, a basic version of the

gossiping time protocol (GTP). In Sec. III, we describe the necessary mean-field theory, and devise a simple analytical model for gossip-based information dissemination as an illustrative example. In Sec. IV we present an analysis of basic GTP using the mean-field convergence result from Sec. III. Sec. V discusses related work. Sec. VI concludes our paper.

## II. 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 [4].

### A. A Generic Gossip Protocol

Figure 1 illustrates the skeleton of a generic gossip-based 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., [1]).

<b>while true do</b>	<b>while true do</b>
<b>wait</b> ( $\Delta t$ time units)	$s_p \leftarrow \text{receive}(\cdot)$ ;
$p \leftarrow \text{RandomPeer}()$ ;	<b>prepare</b> ( $s$ );
<b>prepare</b> ( $s$ );	send $s$ to sender( $s_p$ );
send $s$ to $p$ ;	$s \leftarrow \text{Update}(s, s_p)$ ;
$s_p \leftarrow \text{receive}(\cdot)$ ;	
$s \leftarrow \text{Update}(s, s_p)$ ;	
(a) active thread	(b) passive thread

Fig. 1. The skeleton of a gossip protocol

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 time span of length  $\Delta t$ , and initiates the state exchange (gossip) with it. It does so by sending (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  $\text{Update}(s, s_p)$ ). In response to being contacted by  $A$ , node  $B$  sends (part of) its local state to  $A$  and updates its local state accordingly (method  $\text{Update}(s, s_p)$ ).

Method  $\text{Update}$  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 [5], [6]), a finite list of data items (e.g., news items), called the cache, composes the local state of a node. The generic operation  $\text{prepare}(s)$  in Figure 1 is replaced by an operation  $s \leftarrow \text{RandomItems}()$ . The method  $\text{Update}$  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  $\text{Update}$  (as in [7], [8]) 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, the number of nodes reached in the presence of nodes failures. In probabilistic broadcasting (e.g., [9]), the state of a node is a flag that records whether the node is infected. Method  $\text{Update}$  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 receive the broadcast message. In gossip-based distributed aggregation (e.g. [10]), 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  $\text{Update}$  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 [11] for a thorough survey on gossip applications.

The state exchange between nodes can be implemented as one of the following policies: only the node that initiates a gossip sends state information 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).

### B. Gossiping Time Protocol

Protocols based on epidemic and gossip concepts have found various practical applications [11], including non traditional gossip applications [12], such as gossip-based clock synchronisation. The Gossiping Time Protocol (GTP) [13], [14] is a self-managing gossip time synchronisation protocol for peer-to-peer networks.

The protocol operates in a network of nodes, each of which 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 [1], 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  $\infty$ . 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 [14]: 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_`.

Furthermore, each node may decide to adapt a 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.

Our goal is to show how a mean-field framework can be applied to gossip protocols, on the example of the basic GTP protocol. 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 [13], [14].

## III. 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 [15] but change notations when appropriate and simplify things if possible in the gossip context.

### A. Modelling and Convergence Result

A *discrete-time Markov chain* (DTMC) is a stochastic process  $\{Y(t) \mid t \in \mathbb{N}\}$  that takes values in a countable state

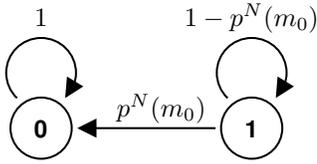


Fig. 2. A single node

space  $S$ . A DTMC obeys the Markov property, that is, the next state is independent of the past, given the present state:

$$\Pr\{Y(t+1) = j \mid Y(0) = i_1, \dots, Y(t) = i_t\} = \Pr\{Y(t+1) = j \mid Y(t) = i_t\}, \quad i_t, j \in S.$$

We consider a system of  $N \in \mathbb{N}$  interacting objects that are identically defined. The object with index  $n \in \{1, \dots, N\}$  is represented by the discrete-time stochastic process  $\{X_n^N(t) \mid t \in \mathbb{N}\}$  which takes values in the set  $S = \{0, \dots, K-1\}$  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 2 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.

The complete system is composed of the  $N$  objects and is, consequently, also described by a discrete-time stochastic process:  $Y^N(t) = (X_1^N(t), \dots, X_N^N(t))$ . 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  $M^N(t) = (M_0(t), \dots, M_{K-1}(t))$  called the *occupancy measure*. Its elements are defined as

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

where  $1_{\{X_n^N(t)=i\}}$  is 1 if  $X_n^N(t) = i$  and 0 otherwise.

Its state space  $S_M^N \subset \mathbb{R}^K$  has

$$|S_M^N| = \binom{K+N-1}{K-1} \quad (1)$$

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  $\mathbf{m} = (m_0, m_1, \dots, m_{K-1}) \in S_M^N$ , 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

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

Its size is

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

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

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

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.

**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

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

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 a fraction of  $m_0$  informed nodes to  $m'_0$  informed nodes, for  $m'_0 \geq m_0$ , equals

$$\binom{m_1 \cdot N}{(m'_0 - m_0) \cdot N} (p^N(m_0))^{(m'_0 - m_0)N} (1 - p^N(m_0))^{m'_1 N},$$

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 finite time  $t \in \mathbb{N}$ . Recall that  $M^N(t)$  is a random

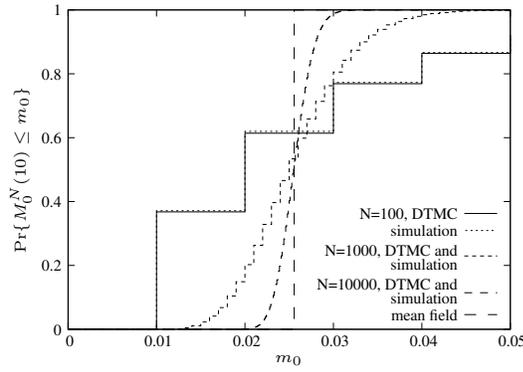


Fig. 3. Distribution of  $M_0^N(10)$  for  $g = 0.1$  and  $M^N(0) = (0.01, 0.99)$

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.

**Example 4** Figure 3 shows the analytically computed (using mean-field analysis) distribution of the fraction of informed nodes at time  $t = 10$ , for gossip probability  $g = 0.1$  and initial occupancy measure  $M^N(0) = (0.01, 0.99)$ . 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 [16]. 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 3.

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 for  $N \rightarrow \infty$

$$P_{i,j}^N(\mathbf{m}) \text{ converges uniformly}^1 \text{ in } \mathbf{m} \text{ to some } P_{i,j}(\mathbf{m}),$$

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  $N \rightarrow \infty$  for each local state  $i$  the fraction  $M_i^N(t)$  of objects in state  $i$  at time  $t$  is known with probability one.

<sup>1</sup>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  $|f_N(x) - f(x)| < \varepsilon$ .

**Theorem 1 (cf. [15])** 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:

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

Define the deterministic process

$$\mu(t+1) = \mu(t) \cdot P(\mu(t)).$$

Then for any  $t \in \mathbb{N}$ ,

$$\lim_{N \rightarrow \infty} M^N(t) = \mu(t), \text{ with probability 1,}$$

that is,  $\mu(t)$  is the deterministic limit occupancy measure for  $N \rightarrow \infty$ .

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

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

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

$$\mu(10) = (0.0256, 0.9744)$$

computed by ten matrix-vector multiplications. It is indicated by the vertical line for  $m_0$  in Figure 3.

## B. 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 – Formal description:* The formal 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 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 to identify 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  $N \rightarrow \infty$  we can 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.

#### IV. A MEAN-FIELD MODEL FOR BASIC GTP

A detailed description of basic GTP can be found in Sec. II-B and in [13], [14]. This corresponds to Step 1 in our methodology. Step 2 and Step 3 are accomplished in the following three subsections (IV-A–C). Section IV-D corresponds to Steps 4 and 5.

##### A. 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. When it is 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 recorded. 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

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

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

##### B. 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  $G : \{0, \dots, H, \infty\} \mapsto \{0, \dots, G_{\max}\}$  that gives the gossip delay  $G(h)$  for any hop count  $h$ .

##### C. 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 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 be  $L$ . 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  $G(0)$ .

$$P_{(g,l,0|g>0),(g-1,L,0)}^N(\mathbf{m}) = 1$$

$$P_{(0,l,0),(G(0),L,0)}^N(\mathbf{m}) = 1.$$

As one can see, time sources act independently of their environment.

*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  $\text{noc}^N(\mathbf{m})$  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

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

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  $\infty$ . 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

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

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

Note that these probabilities are not correct if the new hop count is  $H$ . Since  $H$  subsumes all hop counts of *at least*  $H$ , the probability would be slightly different. We omit this special

case here and in the following presentation, though it has of course been included in the model when computing the results.

It remains the case 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. We again distinguish enforced and optional updates and get the following probabilities.

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

*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

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

The probability of not updating the clock is again composed of three terms: the probability of having a collision, the probability of not being chosen as a peer at all, and the probability of having an interaction with a peer not providing a suitable hop count, as follows:

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

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

#### D. Mean-Field Limits

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

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

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

#### E. Comparison with emulation results

In [14], emulation is used to explore how GTP behaves in practice. For basic GTP a network of 1500 nodes is emulated on a single workstation, using the 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 4(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  $\infty$  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.

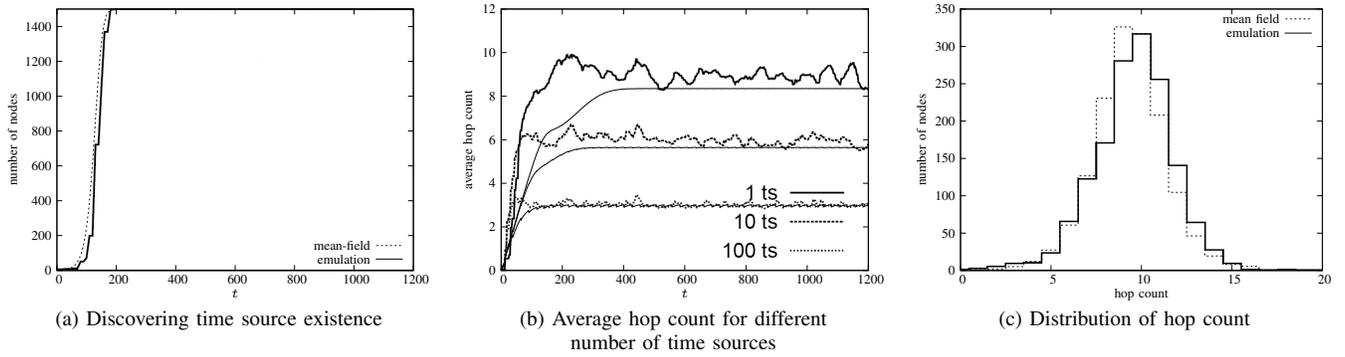


Fig. 4. Comparison with emulation results (includes data from Figures 6.1(a), 6.5(b) and 6.6 in [14])

In Figure 4(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 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. For the emulation we do not know which formula was used to compute the average hop count while there are unsynchronised nodes present. At the beginning, the average hop count increases faster in the emulation, 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.

Figure 4(c) 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 not fully aware of how the distribution was calculated for the emulation, there is a close match between the emulation and the mean-field result.

Figure 4 documents 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 [14] and also evaluate the influence of varying the gossip delay.

### F. More properties of basic GTP

We stick to the scenario of the previous section. Following Figure 4(c) we depict in Figure 5(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 has exactly a given hop count as shown exemplarily 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  $\infty$ . Since this change originates from the time source, hop counts different from  $\infty$  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. Figure 5(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 5(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.

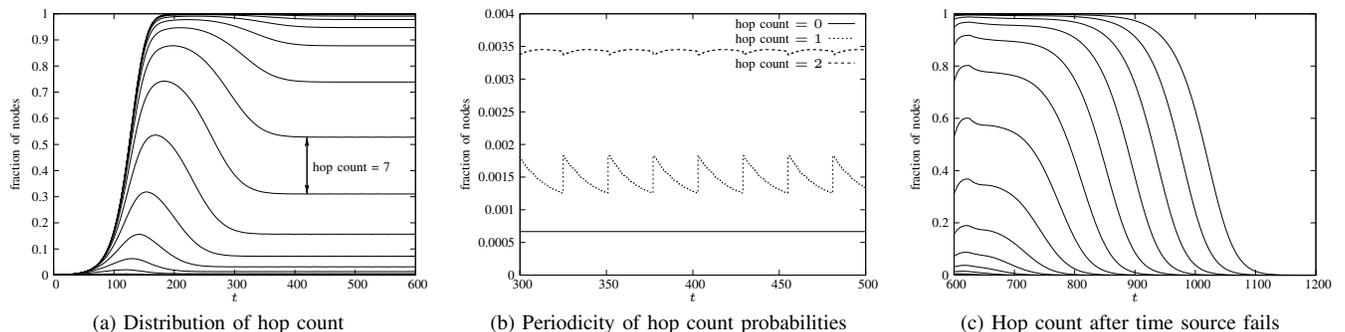


Fig. 5. Hop counts for constant gossip delay (25 seconds)

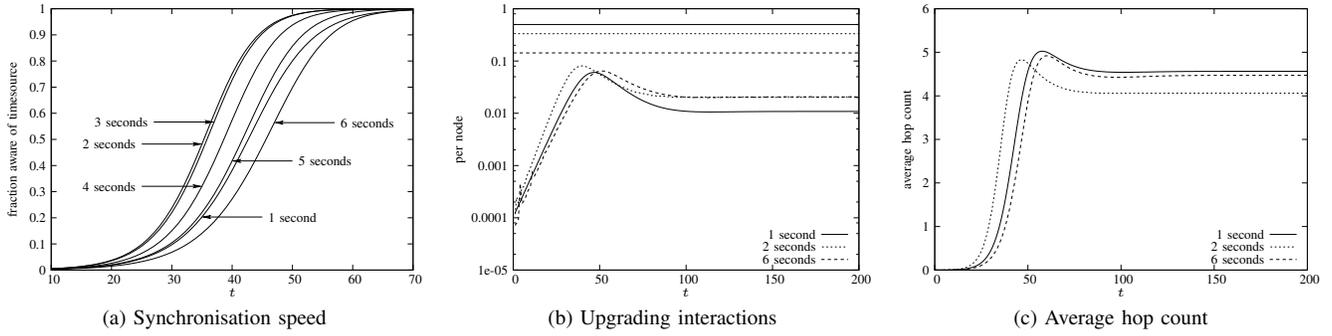


Fig. 6. Varying the static gossip delay

Figure 8(a) 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 clock and hop count to a different value. The hop count might be larger than before if the update has 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.

Finally we want to show the behaviour of the network when the time source fails. We do this by initialising the network with the state after 10 minutes (time  $t = 600$ ), running with a single time source and then redistributing the fraction of nodes being a time source to all other states. Figure 5(c) shows what happens to the hop count distribution in the following 10 minutes. As could be expected, nodes with a low hop count die out over time, leaving all nodes at the chosen maximum hop count of  $H = 15$ .

### G. 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 small gossip delay lead to a slower synchronisation than

a higher gossip delay because of too many interactions and, subsequently, collisions?

Figure 6(a) shows the speed of synchronisation for gossip delays between 1 and 6 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, the synchronisation proceeds slower than for a delay of 2, 3, 4 or 5 seconds. In this case, collisions impede a fast dissemination of the timing information through the network.

Figures 6(b) and 6(c) further substantiate this insight. In 6(b), the upper set of curves depicts the total number of initiated interactions, 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 6(c) documents that also the average hop count for a gossip delay of one is higher than for a gossip delay of two.

### H. Dynamic adaptation of gossip delay

Our model allows 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 [14] we want a node to gossip more often if its time has "bad quality". For our model that 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

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

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  $\infty$  is treated like  $H + 1$ .

We compare three cases: the scenario considered so far with  $G_{\min} = G_{\max} = 25$  seconds, 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 7 depicts both the synchronisation speed (left set of curves) and the average hop count (right set of curves) in the

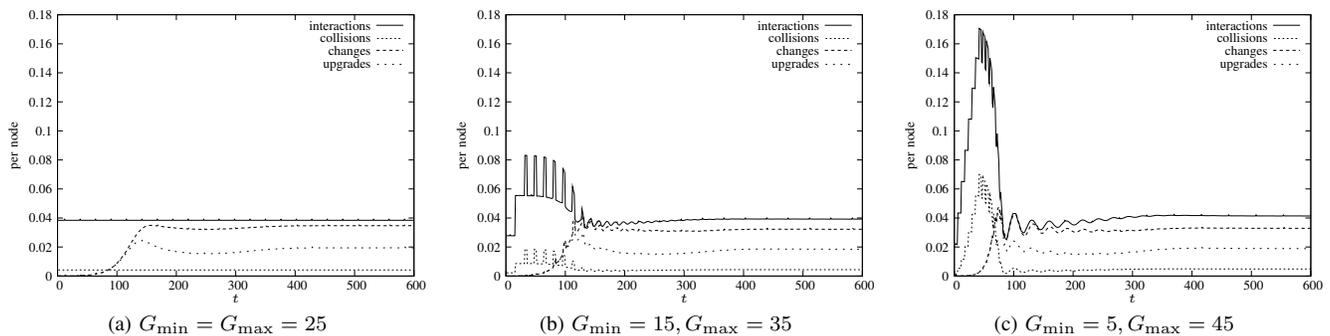


Fig. 8. Interaction activity.

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.

Figure 8 compares the interaction activity of the three scenarios. While for a static gossip delay the number of interactions is independent of the state of the system, 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 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.

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 [14] 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.

## V. RELATED WORK

The notion of “mean-field” is often used in the literature, with different meanings. The mean-field concept was first introduced in physics. 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. In the area of communication networks, mean-field convergence results have been applied in various forms to a variety of case studies, including TCP connections [17], [18], [19], [20], HTTP flows [21], bandwidth sharing [22], transportation networks [23], swarm robotic systems [24], reputation determination [15], queueing networks [25], [26], [27] and Internet congestion control [28].

We are not familiar with prior work dealing with mean-field theory for the evaluation of gossip protocols. Previous work on gossip protocols has used a notion of mean-value and infinite limit (when the number of nodes  $N \rightarrow \infty$ ) to simplify computation for their analysis. Notably in [29], Bonnet studied the evolution of the in-degree distribution of nodes executing the Cyclon protocol [7]. The states of the associated Markov chain represent the fraction of nodes with a specific in-degree distribution. From the designed Markov chain he determined the distribution to which the protocol converges. 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. [30] 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  $N \rightarrow \infty$ ) of the expected delay associated with the broadcasting of a file consisting of a certain amount of packets.

## VI. 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 [31]);

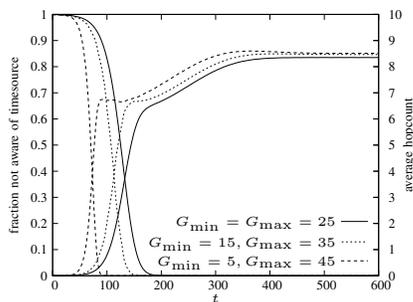


Fig. 7. Synchronisation speed and average hop count.

- 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 [15], the employment of continuous-time models, and steady-state measures [32]. However, the mean-field approach does not allow for the evaluation of a centrally managed network, the separate modelling of one single node or the inclusion of topographic information on the network.

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.

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 [14]. Then we used the mean-field model to derive a large variety of interesting measures, also considering dynamically adjusted gossip delays.

As 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 [32] or by introducing non-determinism using Markov decision processes [33].

## REFERENCES

- [1] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. van Steen, “The peer sampling service: Experimental evaluation of unstructured gossip-based implementations,” in *Proc. ACM/IFIP/USENIX Middleware Conf.*, ser. LNCS, vol. 3231. Springer, 2004, pp. 79–98.
- [2] M. Kwiatkowska, G. Norman, and D. Parker, “Analysis of a gossip protocol in PRISM,” *SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 4, pp. 17–22, 2008.
- [3] R. Bakhshi, L. Cloth, W. Fokkink, and B. R. Haverkort, “Mean-field analysis for the evaluation of gossip protocols,” *SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 3, pp. 31–39, 2008.
- [4] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, “Epidemic algorithms for replicated database maintenance,” in *Proc. ACM Symp. on Principles of Distributed Computing*. ACM Press, 1987, pp. 1–12.
- [5] M. Jelasity, W. Kowalczyk, and M. van Steen, “Newscast computing,” Vrije Universiteit Amsterdam, Tech. Rep. IR-CS-006, 2003.
- [6] D. Gavidia, S. Voulgaris, and 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.
- [7] S. Voulgaris, D. Gavidia, and M. van Steen, “Cyclon: Inexpensive membership management for unstructured P2P overlays,” *J. Network and Syst. Manage.*, vol. 13, no. 2, pp. 197–217, 2005.
- [8] A. Allavena, A. Demers, and J. Hopcroft, “Correctness of a gossip based membership protocol,” in *Proc. ACM Symp. on Principles of Distributed Computing*. ACM Press, 2005, pp. 292–301.
- [9] Q. Zhang and D. Agrawal, “Dynamic probabilistic broadcasting in MANETs,” *J. of Parallel and Distributed Computing*, vol. 65, no. 2, pp. 220–233, 2005.
- [10] M. Jelasity, A. Montresor, and O. Babaoglu, “Gossip-based aggregation in large dynamic networks,” *ACM Trans. Comput. Syst.*, vol. 23, no. 3, pp. 219–252, 2005.
- [11] A.-M. Kermarrec and M. van Steen, “Gossiping in distributed systems,” *SIGOPS Oper. Syst. Rev.*, vol. 41, no. 5, pp. 2–7, 2007.
- [12] P. Costa, V. Gramoli, M. Jelasity, G. P. Jesi, E. Le Merrer, A. Montresor, and L. Querzoni, “Exploring the interdisciplinary connections of gossip-based systems,” *SIGOPS Oper. Syst. Rev.*, vol. 41, no. 5, pp. 51–60, 2007.
- [13] K. Iwanicki, M. van Steen, and S. Voulgaris, “Gossip-based clock synchronization for large decentralized systems,” in *Proc. Workshop on Self-Managed Networks, Systems and Services*, ser. LNCS, vol. 3996. Springer, 2006, pp. 28–42.
- [14] K. Iwanicki, “Gossip-based dissemination of time,” Master’s thesis, Warsaw University and Vrije Universiteit Amsterdam, 2005.
- [15] J.-Y. L. Boudec, D. McDonald, and 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.
- [16] M. Jelasity, A. Montresor, G. P. Jesi, and S. Voulgaris, “PeerSim: A peer-to-peer simulator.” <http://peersim.sourceforge.net/>.
- [17] F. Baccelli, D. McDonald, and J. Reynier, “A mean-field model for multiple TCP connections through a buffer implementing RED,” *Perform. Eval.*, vol. 49, no. 1–4, pp. 77–97, 2002.
- [18] F. Baccelli, A. Chaintreau, D. De Vleeschouwer, and D. McDonald, “HTTP turbulence,” *AMS Networks and Heterogeneous Media*, vol. 1, pp. 1–40, 2006.
- [19] —, “A mean-field analysis of short lived interacting TCP flows,” *SIGMETRICS Perform. Eval. Rev.*, vol. 32, no. 1, pp. 343–354, 2004.
- [20] P. Tinnakornsrisuphap and 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.
- [21] F. Baccelli, D. McDonald, and M. Lelarge, “Metastable regimes for multiplexed TCP flows,” in *Allerton Conf. on Communication, Control, and Computing*, 2004, pp. 1005–1011.
- [22] S. Kumar and L. Massoulié, “Integrating streaming and file-transfer internet traffic: fluid and diffusion approximations,” *Queueing Syst. Theory Appl.*, vol. 55, no. 4, pp. 195–205, 2007.
- [23] L. Afanassieva, S. Popov, and G. Fayolle, “Models for transportation networks,” *J. of Math. Sciences*, vol. 84, no. 3, pp. 1092–1103, 1997.
- [24] A. Martinoli, K. Easton, and W. Agassounon, “Modeling Swarm Robotic Systems: A Case Study in Collaborative Distributed Manipulation,” *Int. Journal of Robotics Research*, vol. 23, no. 4, pp. 415–436, 2004.
- [25] D. Dawson, J. Tang, and Y. Zhao, “Balancing queues by mean field interaction,” *Queueing Syst. Theory & Appl.*, vol. 49, no. 3–4, pp. 335–361, 2005.
- [26] F. Karpelevich, E. Pechersky, and Y. Suhov, “Dobrushin’s approach to queueing network theory,” *J. of Applied Mathematics and Stochastic Analysis*, vol. 9, no. 4, pp. 373–397, 1996.
- [27] N. Vvedenskaya and Y. Suhov, “Dobrushin’s mean-field approximation for a queue with dynamic routing,” *Markov Proc. Rel. Fields*, vol. 3, no. 4, pp. 493–526, 1997.
- [28] W. Kang, F. Kelly, N. Lee, and 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.
- [29] 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.
- [30] I. Stojanovic, M. Sharif, and 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.
- [31] R. Bakhshi, F. Bonnet, W. Fokkink, and B. Haverkort, “Formal analysis techniques for gossiping protocols,” *SIGOPS Oper. Syst. Rev.*, vol. 41, no. 5, pp. 28–36, 2007.
- [32] A. Bobbio, M. Gribaudo, and 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.
- [33] M. Puterman, *Markov Decision Processes*. Wiley, 1994.