Refined Mean Field Analysis of the
Gossip Shuffle Protocol
– extended version –

Nicolas Gast[1], Diego Latella[2], and Mieke Massink[2]

[1] INRIA, France
[2] Consiglio Nazionale delle Ricerche - Istituto di Scienza e Tecnologie
dell'Informazione 'A. Faedo', CNR, Italy

Abstract. Gossip protocols form the basis of many smart collective adaptive systems. They are a class of fully decentralised, simple but robust protocols for the distribution of information throughout large scale networks with hundreds or thousands of nodes. Mean field analysis methods have made it possible to approximate and analyse performance aspects of such large scale protocols in an efficient way. Taking the gossip shuffle protocol as a benchmark, we evaluate a recently developed *refined* mean field approach. We illustrate the gain in accuracy this can provide for the analysis of medium size models analysing two key performance measures. We also show that refined mean field analysis requires special attention to correctly capture the coordination aspects of the gossip shuffle protocol.

1 Introduction and Related Work

Many collective adaptive systems rely on the decentralised distribution of information. Gossip protocols (also known as epidemic or random walk protocols) have been proposed as a paradigm that can provide a stable and reliable method for such decentralised spreading of information [23,6,3,17,9,8,4,2,22]. Gossip protocols are able to scale up to the very large environments that collective adaptive systems are envisioned for. The basic mechanism of information spreading followed by a gossip protocol is that nodes exchange part of the data they keep in their cache with randomly selected peers in pairwise synchronous communications on a regular basis.

Interesting performance aspects of such gossip protocols are the diffusion or replication of a newly inserted fresh data element in a network and the dynamics of network coverage. Diffusion or replication of a data element occurs when nodes exchange the data element in pairwise communication. Two relevant measures are of interest in this case. One is the fraction of the population that has the data element in its cache at a certain point in time (replication). The other concerns

network coverage (coverage), i.e. the fraction of the population of network nodes that have "seen" the data element since its introduction into the network, even if they may no longer have it in their cache due to further exchanges with other peers.

Traditionally, these performance measures have been studied based on simulation models. However, when large populations of nodes are involved, such simulations may be very resource consuming. Recently these protocols have been studied using classic mean field approximation techniques [2,1]. In that classic approach the full stochastic model of a gossip network, i.e. one in which each node is modelled individually, is replaced by a much simpler model in which the pairwise synchronous interactions between individual nodes are replaced by the average effect that all those interactions have on a single node and then the model of this single node is studied in the context of the overall average network behaviour. Of course, the average effects may change over time as nodes may change their local states. This is taken into account in a mean field model by letting the probabilities of interactions possibly depend on the fraction of nodes that are in a particular local state. Compared to traditional simulation methods, mean field approximation techniques scale very well to large populations because these techniques are independent of the exact population size[3]. This method of derivation of a mean field model from a large population of interacting objects relies on what is known as the assumption of "propagation of chaos" (also called "statistical independence" or "decoupling of joint probabilities") [20,7,10,18]. The assumption is based on the fact that when the number of interacting nodes becomes very large, their interactions tend to behave as if they were statistically independent.

However, in reality, we are not always dealing with huge systems, but rather with medium size ones. These are still resource intensive when analysed using simulation and, unfortunately, the classical mean field approximation is less accurate for such medium size systems. For example, in Fig. 1 the results of classical mean field approximation are shown together with a Java based simulation of the protocol for a medium size gossip system with 2500 nodes where initially one node has a new data element that will spread over the network by gossiping. It is easy to see that there is a discrepancy between simulation and classic mean field approximation, both for replication of the data element and for the coverage, even in this not so small system.

In this paper we revisit an analysis of the gossip shuffle protocol using a *refined mean field approximation* for *discrete time population models* that we developed in [12,13], and which was in turn inspired by an earlier result for continuous time population models in [11]. The gossip shuffle protocol was analysed in detail by Bahkshi et al. in [4,2,1] both analytically and by using classic mean field approximation in [2,1] and, more recently, by using on-the-fly mean

---

[3] As long as this size is large enough to obtain a sufficiently accurate approximation. The computational complexity of these techniques *does* depend on the number of local states of an object in a population.
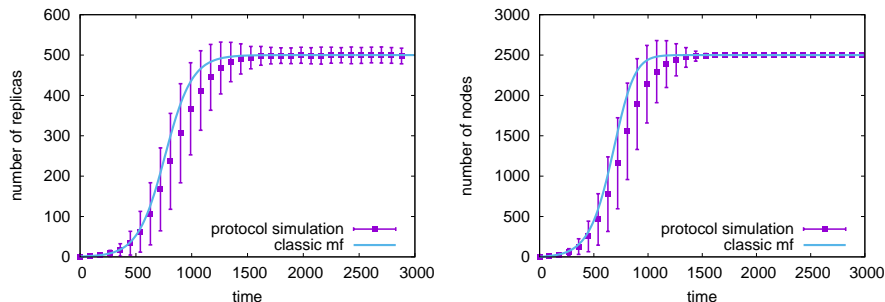
Fig. 1. Replication (left) and Coverage (right) for one new data element in a network with $N = 2500$. Average of 500 simulation runs of the Java simulator [1]. Vertical bars show standard deviation for the simulation.

field discrete time model checking techniques in [19]. The present paper is an extended version of the short paper [14].

Contributions The main contribution of this paper is a novel benchmark (clock-synchronous) DTMC population model of the gossip shuffle protocol analysed using our refined mean field analysis [12,13]. In particular:

- We show that with refined mean field approximation better accuracy can be obtained compared to classical mean field approximation for medium size populations for this gossip protocol, but that this requires a novel model that reflects the synchronisation effects of the pairwise interaction of the original protocol.
- The developed model is parametric in G_{max}, i.e. the number of steps it remains passive in between active interactions with peer gossip nodes.
- The results we obtained are very close both to those of independent Java based simulation from the literature in [2] (taken as "ground truth") and to those of the event simulation of the model itself, but with the advantage that the refined mean field approximation is several orders of magnitude faster to obtain and *independent* of the system size.
- Development of a proof-of-concept implementation in F# of both the classical and the refined mean field techniques and a discrete event simulator used for the analysis of the gossip shuffle protocol [21].

Like classic mean field approaches, the refined approach is computationally non-intensive and *the analysis time is independent of the population size*. The analysis is orders of magnitudes faster than discrete event based simulation. Therefore it is an interesting candidate for being integrated with other analysis approaches such as (on-the-fly) mean field model checking, which is planned in future work. The current study aims at providing further insight in the feasibility of applying the refined mean field approach, that implies the use of symbolic differentiation, on larger benchmark examples and in the possible complications of such an analysis that need to be taken into consideration.

The outline of the paper is as follows. The relevant aspects of the gossip shuffle protocol are briefly recalled in Section 2. The refined mean field approach used in this paper applies to the classical population model of [20,10,18] and is briefly recalled in Section 3. Section 4 presents full and aggregated classical mean field models of the protocol which form the starting point for the novel gossip model suitable for refined mean field approximation presented and analysed in Section 5. Section 6 presents conclusions.

2   Benchmark Gossip Shuffle Protocol

We briefly recall the main aspects of the gossip shuffle protocol described in [15,1,2] that serves as our benchmark. This particular version has been extensively studied by Bahkshi et al., leading to an analytical model of the gossip protocol [3], a classical mean field model [2] and a Java implementation of a simulator for the protocol [2], which makes it a very suitable candidate of a real-world application that allows for the comparison of new results. In the following we briefly recall some main aspects of the shuffle gossip protocol and the Java simulator. Further details can be found in [2,1].

2.1   Informal description

The gossip shuffle protocol distributes data items throughout a network of small devices. Such networks typically consist of a very large collection of nodes. Each node has a limited amount of storage space (called its cache) for the data items. At any instant, gossip nodes are divided into two classes: active and passive nodes. Active nodes can initiate a shuffle, i.e. an exchange of data between two peers, by contacting a passive neighbouring node and exchange part of their data. Such a passive node is selected through an underlying layer[4] that keeps track of which nodes are active or passive.

Each gossip node maintains a finite list of data items in its cache. Both the active node and its passive partner exchange a random subset from their local caches in one atomic peer-to-peer communication session. Given the limited size of the cache, a node may have to discard some items it receives. This is done in such a way that *no information is lost in the network*, i.e. a node discards items selected among those that it has just sent to its peer and does not discard new items it has just received from the peer. Fig. 2 recalls the pseudo code of a generic shuffle protocol (adapted from [1]).

Two main key measures that are of interest for this protocol are the transient aspects of the *replication* of a newly introduced element in the network and that of the *coverage* of the network, i.e. the fraction of network nodes that have seen the new data element when time is passing. These measures depend on a number of characteristics of the network. In the following we use N to denote the

---

[4] This layer is not explicitly modelled. For example, in wireless environments such passive peers may be determined by the radio connectivity between nodes.
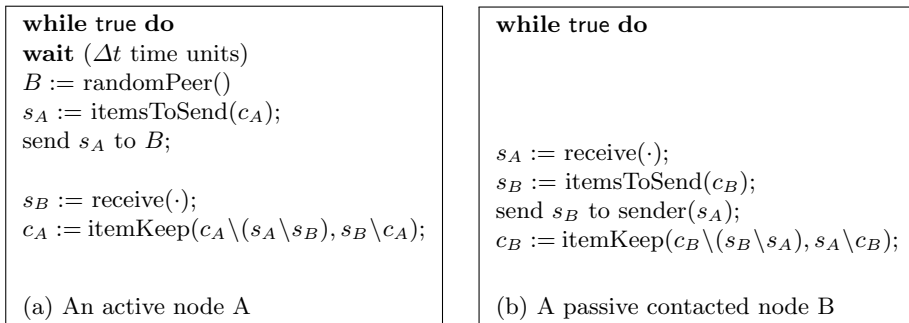
```
while true do
  wait (Δt time units)
  B := randomPeer()
  s_A := itemsToSend(c_A);
  send s_A to B;

  s_B := receive(·);
  c_A := itemKeep(c_A \ (s_A \ s_B), s_B \ c_A);


(a) An active node A
```

```
while true do



  s_A := receive(·);
  s_B := itemsToSend(c_B);
  send s_B to sender(s_A);
  c_B := itemKeep(c_B \ (s_B \ s_A), s_A \ c_B);


(b) A passive contacted node B
```

Fig. 2. Pseudo code of a generic shuffle protocol (adapted from [1]). c_A and s_A denote the cache and selection of active node A. Similarly, c_B and s_B denote those of passive node B. $\Delta t = G_{max}$. The operation 'itemsToSend(c_i)' selects the items to be sent from the cache c_i. The operation 'itemKeep(c,s)' in node A decides which items to keep in the cache (c) removing from the cache those selected for sending (s_A) except those that where received from B (s_B), and adding to those the elements from s_B that were not yet in the cache of A. Similarly for the operation in node B.

size of the network, i.e. the number of gossiping nodes, n to denote the number of *different* data items in the network, c to denote the size of the cache and s to denote the size of the selected items from the cache to be exchanged with a neighbour. In the context of this work, and for comparison with the results presented in [1], the network is assumed to be *fully connected*. We consider a discrete time variant of the protocol with a maximal delay between two subsequent active data-exchanges of a node denoted by G_{max}.

2.2 The gossip Java simulator

To assess the quality of classic mean field approximation results, Bahkshi et al. developed a Java-based implementation of a simulator for the shuffle protocol with which networks of various sizes can be simulated on a single processor [2]. In this paper we also adopt the results produced by this simulator, the source code of which was generously shared with us by the developers, as the "ground truth" with which to compare our own results. This simulator works as follows. It takes the network size N, and the specific size of the storage, c, the number of messages exchanged in each shuffle, s and the total number of different data elements in the network, n. It divides all network nodes into $G_{max} + 1$ different groups, each representing a different value of the gossip delay. Recall that the maximal period between two consecutive contact initiations of any particular network node is G_{max}. The nodes in the group with gossip delay equal to zero are the active nodes, i.e. those that initiate contact with their peers in the current round uniformly at random. If an active node contacts a node that is already in contact with another node, the interaction between all three nodes fails, leading to a collision. At the start of the simulation, a new data item is introduced in

the network (i.e. one different from the n types of data-elements that are already present in the network and that are assumed to be uniformly distributed over the local cash of all network nodes. After each round, the total number of copies of the new data element in the network (replication) and the number of nodes that have seen the data element (coverage) are measured.

3   Background

In the sequel we use theoretical results on discrete time mean field approximation [20,7,12]. We briefly recall the notation and main results in the following. We consider a population model of a system composed of $0 < N \in \mathbb{N}$ identical interacting objects, i.e. a (model of a) system of *size* N. We assume that the set $\{0, \ldots, n-1\}$ of local states of each object is finite; we refer to [12] for a discussion on how to deal with infinite dimensional models. Time is *discrete* and the behaviour of the system is characterised by a (time homogeneous) *discrete time Markov chain* (DTMC) $X^{(N)}(t) = (X_1^{(N)}(t), \ldots, X_N^{(N)}(t))$, where $X_i^{(N)}(t)$ is the state of object i at time t, for $i = 1, \ldots, N$.

The *occupancy measure vector* at time t of the model is the row-vector DTMC $M^{(N)}(t) = (M_0^{(N)}(t), \ldots, M_{n-1}^{(N)}(t))$ where, for $j = 0, \ldots, n-1$, the stochastic variable $M_j^{(N)}(t)$ denotes the *fraction* of objects in state j at time t, over the total population of N objects:

$$M_j^{(N)}(t) = \frac{1}{N} \sum_{i=1}^{N} 1_{\{X_i^{(N)}(t)=j\}}$$

and $1_{\{x=j\}}$ is equal to 1 if $x = j$ and 0 otherwise. At each time step $t \in \mathbb{N}$ each object performs a local transition, possibly changing its state. The transitions of any two objects are assumed to be independent from each other, while the transition probabilities of an object may depend also on $M(t)$, thus, for large N, the probabilistic behaviour of an object is characterised by the one-step transition probability $n \times n$ matrix $\mathbf{K}(m)$, where $\mathbf{K}_{ij}(m)$ is the probability for the object to jump from state i to state j when the occupancy measure vector is $m \in \mathcal{U}^n$; $\mathcal{U}^n$ is the unit simplex of $\mathbb{R}_{\geq 0}^n$, that is $\mathcal{U}^n = \{m \in [0,1]^n \mid \sum_{i=1}^{n} m_i = 1\}$. In this paper, for simplicity, we assume $\mathbf{K}(m)$ to be a continuous function of m that does not depend on N. In the sequel, for reasons of presentation, we provide a graphical specification of the relevant models. The computation of matrix $\mathbf{K}(m)$ from such a model specification is straightforward.

3.1   Discrete Time Classical Mean Field Approximation

Below we recall Theorem 4.1 of [20] on classic mean field approximation, under the simplifying assumptions mentioned above:

Theorem 4.1 of [20] (Convergence to Mean Field) *Assume that the initial occupancy measure $M^{(N)}(0)$ converges almost surely to the*

deterministic limit $\mu(0)$. Define $\mu(t)$ iteratively by (for $t \geq 0$):

$$\mu(t+1) = \mu(t)\,\mathbf{K}(\mu(t)). \qquad (1)$$

Then for any fixed time t, almost surely, $\lim_{N \to \infty} M^{(N)}(t) = \mu(t)$.

The above result thus allows one to use, for *large N*, a *deterministic* approximation μ of the average behaviour of a discrete population model.

3.2   Discrete Time Refined Mean Field Approximation

In [12] we proposed a refined mean field method for *discrete time population models* that has shown to provide a considerably better approximation than classic mean field in the case of population models with a *medium population size N*. This work was inspired by the development of a refined mean field approximation for continuous time population models in [11]. Before recalling the theoretical results for the refined mean field approximation technique for discrete time models we introduce some further basic notation.

$\mathbb{R}_{\geq 0}^n$ denotes the set of n-tuples—i.e. $1 \times n$ matrices—of non-negative real numbers. For $n \times m$ matrix A we let A^T denote its $m \times n$ *transposed* matrix. For function $f : \mathbb{R}^n \to \mathbb{R}^p$ continuous and twice differentiable, let the $p \times n$ (function) matrix $Df(m)$ and the $p \times n \times n$ tensor $D^2 f(m)$ denote its first and second derivatives, respectively: $(Df(m))_{ij} = \frac{\partial f_i(m)}{\partial m_j}$ and $(D^2 f(m))_{ijk} = \frac{\partial^2 f_i(m)}{\partial m_j \partial m_k}$. Let function $\Phi : \mathbb{N} \to \mathcal{U}^n \to \mathcal{U}^n$ be defined as follows:

$$\Phi_0(m) = m; \qquad (\Phi_1(m))_j = \sum_{i=0}^{n-1} m_i \mathbf{K}_{ij}(m); \qquad \Phi_{t+1}(m) = \Phi_1(\Phi_t(m)).$$

Note that $\Phi_1(\Phi_t(m)) = \Phi_t(\Phi_1(m))$ and that, for $\mu(t)$ defined as in Equation (1), we have: $\mu(t+1) = \Phi_1(\mu(t)) = \Phi_{t+1}(\mu(0))$; so, function Φ makes explicit the dependence of $\mu(t)$ on the initial occupancy measure vector m. Suppose function $h : \mathcal{U}^n \to \mathbb{R}_{\geq 0}^p$ models a *measure of interest* over the occupancy measure vectors.

Below we recall Theorem 1 we proved in [12] on *Refined mean-field approximation*:

Theorem 1 of [12] (Refined Mean Field) *Assume that function Φ_1 is twice differentiable with continuous second derivative and that $M^{(N)}(0)$ converges weakly to $\mu(0)$. Let A_t and B_t be respectively the $n \times n$ matrix $A_t = (D\Phi_1)(\mu(t))$ and the $n \times n \times n$ tensor $B_t = (D^2\Phi_1)(\mu(t))$. Then for any continuous and twice differentiable function with continuous second derivative $h : \mathcal{U}^n \to \mathbb{R}_{\geq 0}^p$ we have:*

$$\lim_{N \to \infty} N\mathbb{E}\left[h(M^{(N)}(t)) - h(\Phi_t(M^{(N)}(0)))\right] = Dh(\mu(t))V_t + \frac{1}{2}D^2 h(\mu(t)) \cdot W_t,$$

where V_t is an $n \times 1$ vector and W_t is an $n \times n$ matrix, defined as follows:

$$V_{t+1} = A_t V_t + \tfrac{1}{2} B_t \cdot W_t \text{ and } W_{t+1} = \Gamma(\mu(t)) + A_t W_t A_t^T,$$

with $V_0 = 0$, $W_0 = 0$ and $\Gamma(m)$ is the following $n \times n$ matrix:

$$\Gamma_{jj}(m) = \sum_{i=0}^{n-1} m_i \mathbf{K}_{ij}(m)(1 - \mathbf{K}_{ij}(m))$$

$$\Gamma_{jk}(m) = -\sum_{i=0}^{n-1} m_i \mathbf{K}_{ij}(m)\mathbf{K}_{ik}(m)$$

The following corollary illustrates the relationship between the refined mean field result and the classic convergence theorem:

> **Corollary 1(i) of [12]** *Under the assumptions of* **Theorem 1 of [12]**, *it holds that for any coordinate i and any time-step $t \in \mathbb{N}$*
>
> $$\mathbb{E}\left[M_i^{(N)}(t)\right] = \mu_i(t) + \frac{(V_t)_i}{N} + o\left(\frac{1}{N}\right).$$

In other words, the expected value of the fraction of the objects in local state i of the full stochastic model with population size N at time t, is equal to the classic limit mean field value $\mu_i(t)$ plus a factor that is a constant $(V_t)_i$, calculated as shown in Theorem 1, divided by the population size N plus a residual amount of order $o\left(\frac{1}{N}\right)$. It is easy to see that the larger is N the smaller this additional factor gets. Essentially, the refined mean field takes not only the first moment (the mean) but also the second moment (variance) into consideration in the approximation.

In [12] we have applied this discrete time refined mean field approximation on a number of examples ranging from the well-known epidemic model SEIR to wireless networks. It was shown that the approach works well under the assumption that the models have a unique fixed point and *exponentially stable behaviour*, i.e. possible oscillations in the behaviour of the system, due to a finite input, will die out at an exponential rate. Here we investigate its application to the more complex gossip shuffle protocol.

A proof-of-concept implementation of both the classical and the refined mean field techniques and a discrete event simulator has been developed by one of the authors of the present paper in F# using the DiffSharp package [5] for symbolic differentiation. The results in this paper have been obtained using this implementation which can be found at [21].

4   Gossip Shuffling Protocol Mean Field Model

Following the classic discrete time mean field approximation technique [20,1,2] the behaviour of an individual gossip node can be described based on its local state and the current occupancy measure vector. This exploits what is known as the "decoupling principle", i.e. in the limit for N going to infinity, the evolution of each individual object is assumed to be *stochastically independent* from other specific objects – except through dependence on the global occupancy measure – even in the presence of explicit cooperation (i.e. synchronisation) between objects [20,7,16]. Such a model of an individual node can then be used to analyse

global properties of the network such as the replication and coverage measures that are relevant in this case study.

Without going into full detail[5], the mean field models proposed in the work by Bahkshi et al. [2] consider a gossip network as consisting of active and passive nodes that possess, or do not possess, the specific data element in their cache. This is illustrated in Fig. 3 (left), where the local states of a single node are shown. States in which the node actively looks for a gossip peer are red, those in which it passively receives requests are blue. States in which the node has the data-element in its cache are labelled by D_i, those in which it does not are labelled by O_i. Transitions between states occur with certain probabilities, which depend on the global occupancy measure and the conditional probabilities of pairwise node interaction, shown in Fig. 3 (right), under the assumption of a uniform distribution of data items over the local storages of all nodes. $P(A'B'|AB)$ denotes the conditional probability of the state of an active-passive pair AB to have state $A'B'$ after their interaction, where $A, B, A', B' \in \{O, D\}$.
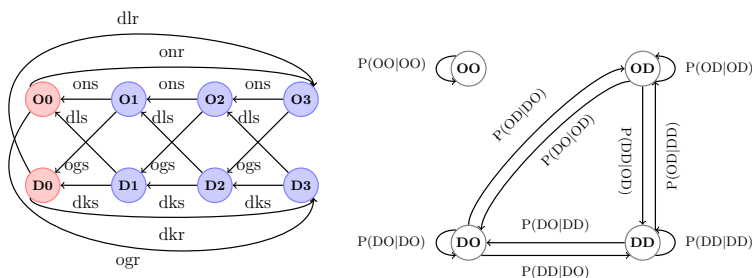

Fig. 3. Left: Push-pull gossip model of individual gossip node with rounds of length 3 (i.e. $G_{max} = 3$). Active states are red, passive ones blue. Model for replication. Right: Transition diagram of conditional probabilities pairwise interaction between gossip nodes. D: data element in cache; O: data element not in cache.

The conditional probabilities[6] can be expressed in terms of n (number of different data elements), c (size of the cache) and s (number of selected elements for exchange), as follows:

---

[5] More details can be found in the Appendix, which will not be part of this paper.
[6] See [1,2] for further details on this pairwise communication probabilities.

$$P(OD|DO) = P(DO|OD) = \frac{s}{c} * \frac{n-c}{n-s}$$
$$P(OD|OD) = P(DO|DO) = \frac{c-s}{c}$$
$$P(DD|OD) = P(DD|DO) = \frac{s}{c} * \frac{c-s}{n-s}$$
$$P(OD|DD) = P(DO|DD) = \frac{s}{c} * \frac{c-s}{c} * \frac{n-c}{n-s}$$
$$P(DD|DD) \qquad = 1.0 - 2.0 * \frac{s}{c} * \frac{c-s}{c} * \frac{n-c}{n-s}$$
$$P(OO|OO) \qquad = 1.0$$

This mean field model can be further simplified, leading to a model that is parametric in G_{max}, by aggregating the O-states and the D-states, respectively. This uses the experimental observation that when, in the initial state, the O-states all have the same occupancy measure, and all the D-states have the same occupancy measure, this situation remains so when time evolves[7]. This observation is illustrated in Fig. 4 for a network with 2500 nodes and $G_{max} = 3$, with 10 nodes in each O-state and 615 nodes in each D-state initially.
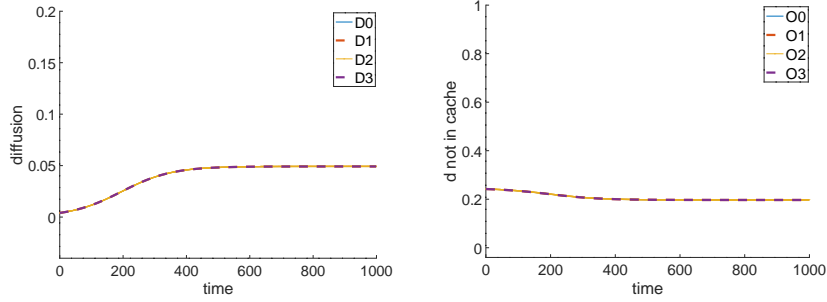

Fig. 4. Diffusion of d-element in the network for $N = 2500$ with initially 615 nodes in each O-state and 10 nodes in each D-state. Occupancy measure of D-states (left) and of O-states (right).

The simplified aggregated mean field model is shown in Fig. 5 (left) for the analysis of the replication, and on the right for the analysis of coverage. The latter shows an additional state (I). This models the state in which the node does not have the data-element in its cache currently and also has never had it before. The O-state in this model represents the fact that it does not have the data element at the moment, but that is has seen it previously (i.e. the node is already covered).

The transition probability functions in the three-state model of Fig. 5, with states O, D and I are defined as follows, for $m = (m_O, m_D) \in \mathcal{U}^2$:

- from O to D: $\mathsf{get}\,(m) = \frac{G_{max}}{G_{max}+1}(\mathsf{ogs}'\,(m)) + \frac{1}{G_{max}+1}(\mathsf{ogr}'\,(m))$
- from D to O: $\mathsf{loose}\,(m) = \frac{G_{max}}{G_{max}+1}(\mathsf{dls}'\,(m)) + \frac{1}{G_{max}+1}(\mathsf{dlr}'\,(m))$

---

[7] Note that we do *not* assume that the occupancy measure of an O-state is equal to that of a D-state.
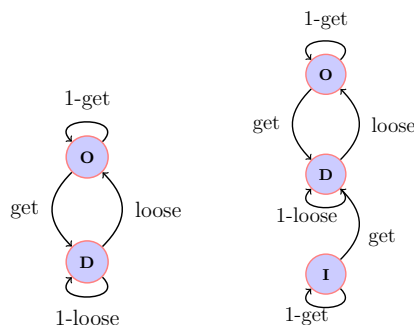
Fig. 5. Two-state (left) and three-state (right) aggregate push-pull gossip model of an individual gossip node with rounds of length G_{max}.

– from I to D: get as above

where

$\mathsf{ogs}'(m) = \frac{1}{G_{max}+1} m_D (P(OD|DO) + P(DD|DO))\mathsf{noc}$

$\mathsf{dls}'(m) = \frac{1}{G_{max}+1} ((m_O + m_I)P(DO|OD) + m_D P(DO|DD))\mathsf{noc}$

$\mathsf{ogr}'(m) = \frac{G_{max}}{G_{max}+1} m_D (P(DO|OD) + P(DD|OD))\mathsf{noc}$

$\mathsf{dlr}'(m) = \frac{G_{max}}{G_{max}+1} ((m_O + m_I)P(OD|DO) + m_D P(DO|DD))\mathsf{noc}$

where noc is the no-collision probability, which, in the aggregated models, is equal to $e^{-2*(1/(G_{max}+1))}$; note that $1/(G_{max}+1)$ is the fraction of active nodes in the network at any time instant. This is derived from [1], where it is shown that in the limit for N to infinity, the probability of no collision is given by $e^{-2*(\mathsf{frc}\,(O0)+\mathsf{frc}\,(D0))}$ where the sum $\mathsf{frc}\,(O0) + \mathsf{frc}\,(D0)$ denotes the fraction of active nodes in the network at any time. In the aggregated model this amounts to $1/(G_{max}+1)$. In this model the number of replications of the data element in the network corresponds to the number of nodes that are in state D. The coverage of the network is given by the number of nodes that are in state D or state O. The definitions of the transition probabilities for the two-state model are similar, but with m_I equal to zero. With the two state model only the number replications can be analysed.

For very large systems both models show a surprisingly good correspondence between the Java simulation results and the classic mean field approximation. For N=25,000 the curves for both measures essentially overlap (see [1,2]). For N=2,500, with initially one node in state D and all other nodes in state I, for $G_{max} = 9$, the results for replication and coverage are shown in Fig. 1. For that system size already some differences can be observed, and, even though they are not huge, there is a considerable difference in the time at which network coverage seems to be reached. The Java simulation (average of 500 runs) indicates that this happened close to time 1500, whereas the mean field indicates a time well before that, just before time 1000, even though the mean field approximation

is still just within the standard deviation of the simulation runs. In the next section we illustrate what results can be obtained with the refined mean field approximation and we also motivate why, in the general case, this requires a more detailed mean field model.

5   Refined Mean Field Approximation of the Gossip Shuffle Protocol

The mean field models of the gossip shuffle protocol in the previous section were based on the principle of decoupling of joint probabilities [20,7] based on a careful study of the pairwise probabilities of the various possible outcomes of a shuffle between two gossip nodes (as in [1]). In our previous work on refined mean field approximation we have shown for a number of other models that this approximation technique can provide an increased accuracy w.r.t. classical mean field and that there is also a close correspondence between the simulation of the mean field model and the refined approximation [12,13]. However, simulation of the *mean field model*[8] of Fig. 3 for a small network of size N=120, with $G_{max} = 3$, with initially 29 nodes in each O-state and one in each D-state, shows that in many simulation runs the system completely looses the introduced data-element. In other words, no gossip node in the network has the element in its cache at a certain point in time. This is clearly in contrast with the properties of the gossip protocol itself. The refined mean field approximation is also sensitive to this aspect of the model behaviour as can be observed in Fig. 6. Similar observations can be made for the aggregated 3-state model of Fig. 5.
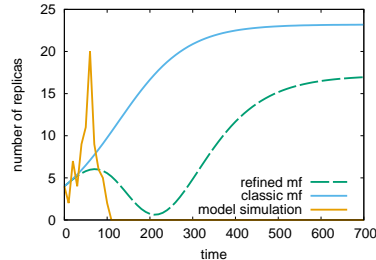

Fig. 6.   Replication of data element in the network for $N = 120$, $G_{max} = 3$, with initially 29 nodes in each O-state and 1 node in each D-state showing a single simulation trace of the model of Fig. 3 (left)in which the data element gets lost from the network. The figure also shows the classic mean field (blue) and refined mean field (green) results.

In the following we propose a more detailed mean field model in which (1) the system can never completely loose the inserted data element and (2) the model

---

[8] We really intend the simulation of the *model* here, and *not* the Java simulation of the protocol.

reflects more explicitly the *effects* of the pairwise interaction and synchronisation between nodes. Note the emphasis on *effects* of node synchronisation because we still are aiming at a model that respects the decoupling principle for its use in a mean field setting. What we really aim at is to distinguish the effects of a node getting a data element through *exchanging* it with another node–in which case the total number of replicas of the data element in the system remains the same– or through *replication*, i.e. the other node retains its copy of the data element and the global number of the data element in the system increases by one.

With reference to Fig. 7, for what concerns point (1) above, we introduce a specific state, PD, to the model representing that there always is a gossip node in the network that possesses the data element.

To address point (2), we introduce two more states, FD and LD, to distinguish between the effect of interactions between gossip nodes. State FD represents the fact that the gossip node received the data element for the *first time* via an *exchange* of the data element with another node. State LD also represents the fact that the node received the data element via an *exchange*, but that it had already seen the data element in the past. So in both cases, the data element is simply exchanged, i.e. one node gives it to the other, and the total number of gossip nodes that possess the data element is not changed by such an interaction. Note that modelling the effect of an exchange of the data element between two nodes in this way also means that we can retrieve the total number of gossip nodes in the system that do *not* possess the data element as *the sum* of the nodes that are in states FD, LD, I and O. This is so because we know that for each node in state FD (LD, resp.) there is a node in the network that just lost its data element in the synchronous shuffle with our current node. We will make use of this in the probability functions associated with the transitions between nodes.

A gossip node can also get involved in an interaction in which the data element is replicated, i.e. a node gives it to another one but also retains a copy itself. Note that this can happen both in case the node that receives the data element does not possess the data element and when it does possess it. This situation is modelled by state D and represents the fact that the interaction has the effect that the total number of nodes in the network that possess the data element increases (by one).

A third case exists where two nodes, both possessing the data element, interact and one of them looses its copy. In that case the overall number of copies of the data element in the network is reduced by one. Note that the gossip protocol does not allow that *both* copies get lost in such an interaction. Moreover, if there is only a single node left in the network with a copy of the data element this copy cannot get lost because this node cannot interact with another node having the data element.

To distinguish the various kinds of interactions mentioned above we refine the transition probability functions introduced on page 10. In particular, we split the probability functions get and loose into two distinct parts, get_rep and get_exc for the get function to model data element replication and exchange, respectively,

and likewise for the loose function as follows, where the appropriate conditional probabilities are used:

$$\begin{aligned}
\mathsf{get\_exc}\,(m) &= 2 * \tfrac{G_{max}}{(G_{max}+1)^2}(m_D + m_{PD})P(OD|DO)\mathsf{noc} \\
\mathsf{get\_rep}\,(m) &= 2 * \tfrac{G_{max}}{(G_{max}+1)^2}(m_D + m_{PD})P(DD|DO)\mathsf{noc}
\end{aligned}$$

$$\begin{aligned}
\mathsf{loose\_exc}\,(m) &= 2 * \tfrac{G_{max}}{(G_{max}+1)^2}(m_O + m_I + m_{LD} + m_{FD})P(OD|DO)\mathsf{noc} \\
\mathsf{loose\_rep}\,(m) &= 2 * \tfrac{G_{max}}{(G_{max}+1)^2}(m_D + m_{PD})P(DO|DD)\mathsf{noc}
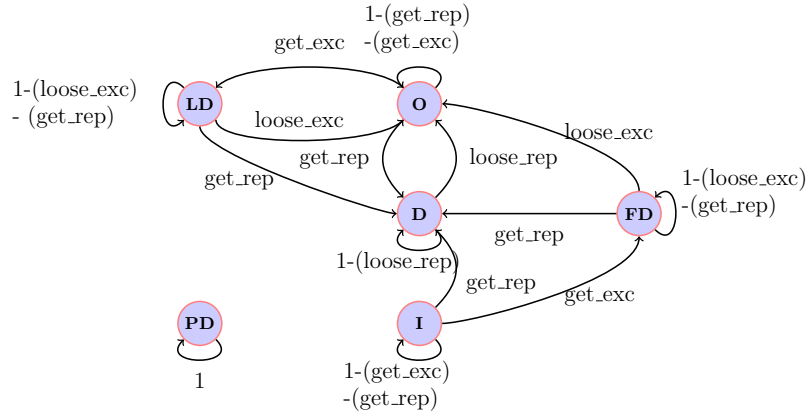\end{aligned}$$

Fig. 7. Six-state model of an individual gossip node with rounds of length G_{max}.

Fig. 8 shows the replication as sum of the number of nodes in states D and PD and the coverage as the sum of the number of nodes in D, PD, FD, LD and O[9] for a network with $N = 100$, $n = 500$, $c = 100$ and $s = 50$ with initially one node in state PD and all the others in state I. Besides the classic and refined mean field approximations for the model in Fig. 7 and the Java simulation results of the actual shuffle protocol, Fig. 8 also shows the average of the model simulation. In particular, note the good approximation of the simulation results (both the Java simulation and the model simulation) by the refined mean field even in this very small network. This holds both for the diffusion of the replicas and for the coverage. Similarly good results have been found for a system with N=2,500 shown in Fig. 9, also in the case in which there is only a single data element in the system initially. An indication of the (non-optimised) performance of the analysis for producing the results in Fig. 9 is: 0.543s (classic mean field); 25.459s (refined mean field); 7m 1.389s (fast model simulation [20], 500 runs); 3h 42m 41.459s (Java simulation, 500 runs) on a MacBook Pro, Intel i7, 16GB. Recall

---

[9] For the refined mean field this means the application of Thm. 1 with $h(m) = m_D + m_{PD}$ (replication) and $h(m) = m_D + m_{PD} + m_{FD} + m_{LD} + m_O$ (coverage), repectively.
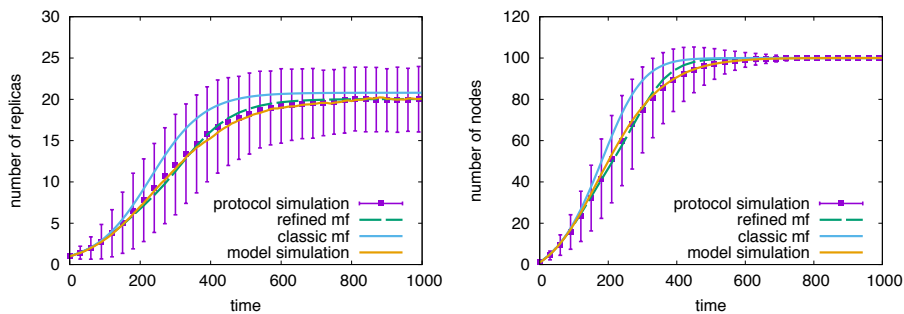
Fig. 8. Replication (left) and network coverage (right) of the data element in the network for $N = 100$ with initially 99 nodes in the I-state and 1 node in the PD-state for $G_{max} = 3$. Average of 500 simulation runs of both the model and Java simulations. Vertical bars show standard deviation for the Java simulation.

that the mean field and refined mean field analyses times are *independent* of the size of the system and, as can be seen, several orders of magnitude faster than traditional event simulation approaches.

6   Conclusion

Gossip protocols play an important role in the design of collective adaptive systems providing a basic, but robust and scalable, mechanism of information spreading in very large networks. Therefore they also form an interesting bench-mark application for the analysis of scalable verification techniques. We have developed a new mean field model for the shuffle gossip protocol with which more accurate approximations for medium size gossip protocols can be obtained via refined mean field approximation techniques. This model respects key aspects of the protocol such as the effects of different kinds of interactions and the fact that a new data element cannot be lost by the system as a whole.

Good approximation of medium size systems is of interest for several reasons. First of all, many practical systems consist of many, but not a huge number, of components. However, even in case of medium size systems, simulation is still a resource consuming effort and in that case a refined mean field approximation can
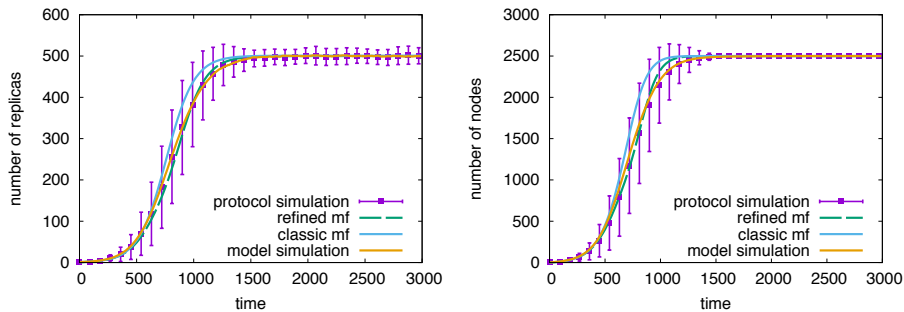
Fig. 9. Replication (left) and network coverage (right) of data element for $N = 2500$ with initially 2499 nodes in I and 1 in PD, for $G_{max} = 9$. Average of 500 simulation runs for both model and Java simulations. Vertical bars show standard deviation for the Java simulation.

provide fast but accurate approximations. Furthermore, we expect that refined mean field approximation can also be of use when analysing systems in which objects are mobile and move through physical space. The uneven distribution of objects over partitions of such a space requires a mean field approximation that is accurate also for those partitions with relatively few objects.

7   Acknowledgements

References

1. Bakhshi, R.: Gossiping Models – Formal Analysis of Epidemic Protocols. Ph.D. thesis, Vrije Universiteit Amsterdam (January 2011), http://www.cs.vu.nl/en/Images/Gossiping_Models_van_Rena_Bakhshi_tcm210-256906.pdf
2. Bakhshi, R., Cloth, L., Fokkink, W., Haverkort, B.R.: Mean-field framework for performance evaluation of push-pull gossip protocols. Perform. Eval. 68(2), 157–179 (2011), https://doi.org/10.1016/j.peva.2010.08.025

3. Bakhshi, R., Gavidia, D., Fokkink, W., van Steen, M.: An analytical model of information dissemination for a gossip-based protocol. Computer Networks 53(13), 2288–2303 (2009), `https://doi.org/10.1016/j.comnet.2009.03.017`
4. Bakhshi, R., Gavidia, D., Fokkink, W., van Steen, M.: A modeling framework for gossip-based information spread. In: Eighth International Conference on Quantitative Evaluation of Systems, QEST 2011, Aachen, Germany, 5-8 September, 2011. pp. 245–254. IEEE Computer Society (2011), `https://doi.org/10.1109/QEST.2011.39`
5. Baydin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M.: Automatic differentiation in machine learning: a survey. Journal of Machine Learning Research 18, 153:1–153:43 (2018), `http://jmlr.org/papers/v18/17-468.html`
6. Birman, K.: The promise, and limitations, of gossip protocols. Operating Systems Review 41(5), 8–13 (2007), `https://doi.org/10.1145/1317379.1317382`
7. Bortolussi, L., Hillston, J., Latella, D., Massink, M.: Continuous approximation of collective system behaviour: A tutorial. Perform. Eval. 70(5), 317–349 (2013)
8. Frei, R., Serugendo, G.D.M.: Advances in complexity engineering. International Journal of Bio-Inspired Computation 3(4), 199–212 (2011), `https://doi.org/10.1504/IJBIC.2011.041144`
9. Frei, R., Serugendo, G.D.M.: Concepts in complexity engineering. International Journal of Bio-Inspired Computation 3(2), 123–139 (2011), `https://doi.org/10.1504/IJBIC.2011.039911`
10. Gast, N., Gaujal, B.: A mean field approach for optimization in discrete time. Discrete Event Dynamic Systems 21(1), 63–101 (2011), `https://doi.org/10.1007/s10626-010-0094-3`
11. Gast, N., Houdt, B.V.: A refined mean field approximation. Proceedings of the ACM on Measurement and Analysis of Computing Systems 1(2), 33:1–33:28 (2017), `https://doi.org/10.1145/3154491`
12. Gast, N., Latella, D., Massink, M.: A refined mean field approximation of synchronous discrete-time population models. Perform. Eval. 126, 1–21 (2018), `https://doi.org/10.1016/j.peva.2018.05.002`
13. Gast, N., Latella, D., Massink, M.: A refined mean field approximation for synchronous population processes. In: Workshop on MAthematical performance Modeling and Analysis (MAMA 2018). pp. 30–32. ACM SIGMETRICS Performance Evaluation Review, ACM (2019)
14. Gast, N., Latella, D., Massink, M.: Refined mean field analysis: the gossip shuffle protocol revisited. In: Bocchi, L., Bliudze, S. (eds.) Coordination Models and Languages - 22th IFIP WG 6.1 International Conference, COORDINATION 2020. LNCS, Springer (2020), short paper, to appear.
15. Gavidia, D., Voulgaris, S., van Steen, M.: A gossip-based distributed news service for wireless mesh networks. In: Conf. on Wireless On demand Network Systems and Services (WONS). pp. 59–67. IEEE Computer Society (2006)
16. Gottlieb, A.D.: Markov Transitions and the Propagation of Chaos. Office of scientific and Technical Information (OSTI), U.S. Department of Energy. (2013), also available at: https://arxiv.org/abs/math/0001076
17. Jelasity, M.: Gossip. In: Serugendo, G.D.M., Gleizes, M.P., Karageorgos, A. (eds.) Self-organising Software - From Natural to Artificial Adaptation, pp. 139–162. Natural Computing Series, Springer (2011), `https://doi.org/10.1007/978-3-642-17348-6_7`
18. Latella, D., Loreti, M., Massink, M.: On-the-fly PCTL fast mean-field approximated model-checking for self-organising coordination. Sci. Comput. Program. 110, 23–50 (2015), `https://doi.org/10.1016/j.scico.2015.06.009`

19. Latella, D., Loreti, M., Massink, M.: Flyfast: A mean field model checker. In: Legay, A., Margaria, T. (eds.) Tools and Algorithms for the Construction and Analysis of Systems - 23rd International Conference, TACAS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10206, pp. 303–309 (2017), `https://doi.org/10.1007/978-3-662-54580-5_18`
20. Le Boudec, J., McDonald, D.D., Mundinger, J.: A generic mean field convergence result for systems of interacting objects. In: Fourth International Conference on the Quantitative Evaluaiton of Systems (QEST 2007), 17-19 September 2007, Edinburgh, Scotland, UK. pp. 3–18. IEEE Computer Society (2007)
21. Massink, M.: Refined mean field F# implementation and gossip shuffle model, `https://github.com/mimass/RefinedMF`
22. Pianini, D., Beal, J., Viroli, M.: Improving gossip dynamics through overlapping replicates. In: Lluch-Lafuente, A., Proença, J. (eds.) Coordination Models and Languages - 18th IFIP WG 6.1 International Conference, COORDINATION 2016, Held as Part of the 11th International Federated Conference on Distributed Computing Techniques, DisCoTec 2016, Heraklion, Crete, Greece, June 6-9, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9686, pp. 192–207. Springer (2016), `https://doi.org/10.1007/978-3-319-39519-7_12`
23. Voulgaris, S., Jelasity, M., van Steen, M.: A robust and scalable peer-to-peer gossiping protocol. In: Moro, G., Sartori, C., Singh, M.P. (eds.) Agents and Peer-to-Peer Computing, Second International Workshop, AP2PC 2003, Melbourne, Australia, July 14, 2003, Revised and Invited Papers. Lecture Notes in Computer Science, vol. 2872, pp. 47–58. Springer (2003), `https://doi.org/10.1007/978-3-540-25840-7_6`

8   Appendix: Detailed Models and Proofs

8.1   Details for gossip model in Fig. 3

Fig. 3 (left) shows the states and transitions of a single gossip node where $G_{max} = 3$. The red states, $D0$ and $O0$, denote states in which the gossip node is *active*, i.e. it can initiate an exchange of local information with a passive node; in $D0$ (resp. $O0$) the node has (resp. does not have) the data element in its local cache. The blue states denote states in which the node is passive and is available for data exchange with an active node when contacted by the latter. The number in the node-labels denotes the value, ranging from 0 to 3, of the current gossip delay g before the node becomes active again. The D/O convention w.r.t. having the data element applies also to the states where the node is passive. The transition labels in Fig. 3 (left) are shorthands for transition probability functions. The latter depend on the occupancy measure vector. Their definition makes use of the conditional probabilities shown in Fig. 3 (right). Furthermore, we recall from [1] that there is a small probability that collision occurs in the communication between two nodes. This happens when a gossip partner is selected that is already involved in a shuffle with another node. In the limit for N to infinity, the value of the probability of no collision is given by $e^{-2*(\mathsf{frc}\,(O0)+\mathsf{frc}\,(D0))}$ where $\mathsf{frc}\,(O0)+$

$\mathsf{frc}\,(D0)$ denotes the fraction of active nodes in the network at any time, i.e. summing active nodes that have the d-element and those that do not.

In the definition of the transition probability functions, we also make use of the following observation that greatly simplifies the definitions. Note that this gossip model is a clock-synchronous model and that each node gets active every G_{max} time steps. This means that in every step, if the node is in state Di (or Oi) then in the next step it leaves this state with probability 1.0 to move one step closer towards the active state $D0$, or, if it was active, it moves to $D3$ (or $O3$) modelling a reset of the time-to-activation; similarly for Oi states. In other words, for all time steps t it holds that:

$$\mathsf{dks}(\mu(t)) + \mathsf{dls}(\mu(t)) = 1 \text{ and } \mathsf{ons}(\mu(t)) + \mathsf{ogs}(\mu(t)) = 1$$

Similarly for the reset probabilities. The proofs can be found in Sect. 8.7 of this Appendix.

The transition probability functions that concern the D-states are defined as shown below. Note that here and in the remainder of the Appendix we also use Currying notation for notational simplicity. In the sequel m denotes the occupancy measure *vector*. Its components are indicated by m_{O1}, m_{O2} and so on.

$$
\begin{aligned}
\mathsf{d\_loss\_reset}\ m\ &= (m_{O1} + m_{O2} + m_{O3}) * P(OD|DO) * (\mathsf{noc}\ )+\\
&\quad\ (m_{D1} + m_{D2} + m_{D3}) * P(OD|DD) * (\mathsf{noc}\ )\\[4pt]
\mathsf{d\_loss\_step}\ m\ &= m_{O0} * P(DO|OD) * (\mathsf{noc}\ ) + m_{D0} * P(DO|DD) * (\mathsf{noc}\ )\\[4pt]
\mathsf{d\_keep\_reset}\ m &= 1.0 - (\mathsf{d\_loss\_reset}\ m)\\[4pt]
\mathsf{d\_keep\_step}\ m\ &= 1.0 - (\mathsf{d\_loss\_step}\ m)
\end{aligned}
$$

Function d_loss_reset corresponds to the transition dlr in Fig. 3, and so on.

The transition probability functions concerning the O-states are:

$$
\begin{aligned}
\mathsf{o\_getd\_reset}\ m &= (m_{D1} + m_{D2} + m_{D3}) * (P(DO|OD) + P(DD|OD)) * (\mathsf{noc}\ )\\[4pt]
\mathsf{o\_getd\_step}\ m\ &= m_{D0} * (P(OD|DO) + P(DD|DO)) * (\mathsf{noc}\ )\\[4pt]
\mathsf{o\_nod\_reset}\ m\ &= 1.0 - (\mathsf{o\_getd\_reset}\ m)\\[4pt]
\mathsf{o\_nod\_step}\ m\ \ &= 1.0 - (\mathsf{o\_getd\_step}\ m)
\end{aligned}
$$

Defining a 8×8 matrix $\mathbf{K}$ with indexes i, j in $\{0, \cdots, 7\}$ then we can define the gossip model as follows for the non-zero elements of $\mathbf{K}$:

$$
\begin{aligned}
K_{0,3} &= 1.0 - (\textsf{o\_getd\_reset } m) \\
K_{0,7} &= \textsf{o\_getd\_reset } m \\
K_{1,0} = K_{2,1} = K_{3,2} &= 1.0 - (\textsf{o\_getd\_step } m) \\
K_{1,4} = K_{2,5} = K_{3,6} &= \textsf{o\_getd\_step } m \\
\\
K_{4,3} &= \textsf{d\_loss\_reset } m \\
K_{4,7} &= 1.0 - \textsf{d\_loss\_reset } m \\
K_{5,0} = K_{6,1} = K_{7,2} &= \textsf{d\_loss\_step } m \\
K_{5,4} = K_{6,5} = K_{7,6} &= 1.0 - \textsf{d\_loss\_step } m
\end{aligned}
$$

This leads a set of eight difference equations for the model recalling that the occupancy vector m at time $t+1$ is $m(t+1) = m(t) * \mathbf{K}(m(t))$. These equations are shown in full in the next section.

8.2   Difference equations for the gossip model in Fig. 3

We obtain the following set of difference equations[10] for the O-states and the D-states of the model, representing the occupancy measure of state Oi by m_{Oi}, and Di by m_{Di}, for $i \in \{0, \cdots, 3\}$.

$$
\begin{aligned}
m_{O0} &= m_{O1} - m_{O1} * (\textsf{o\_getd\_step } m) + m_{D1} * (\textsf{d\_loss\_step } m) \\
m_{O1} &= m_{O2} - m_{O2} * (\textsf{o\_getd\_step } m) + m_{D2} * (\textsf{d\_loss\_step } m) \\
m_{O2} &= m_{O3} - m_{O3} * (\textsf{o\_getd\_step } m) + m_{D3} * (\textsf{d\_loss\_step } m) \\
m_{O3} &= m_{O0} - m_{O0} * (\textsf{o\_getd\_reset } m) + m_{D0} * (\textsf{d\_loss\_reset } m) \\
\\
m_{D0} &= m_{O1} * (\textsf{o\_getd\_step } m) + m_{D1} - m_{D1} * (\textsf{d\_loss\_step } m) \\
m_{D1} &= m_{O2} * (\textsf{o\_getd\_step } m) + m_{D2} - m_{D2} * (\textsf{d\_loss\_step } m) \\
m_{D2} &= m_{O3} * (\textsf{o\_getd\_step } m) + m_{D3} - m_{D3} * (\textsf{d\_loss\_step } m) \\
m_{D3} &= m_{O0} * (\textsf{o\_getd\_reset } m) + m_{D0} - m_{D0} * (\textsf{d\_loss\_reset } m)
\end{aligned}
$$

8.3   Classic Mean Field Model: Coverage.

Network coverage at time t denotes the fraction of the gossip nodes that have seen the data element at any point in time t', with $t_0 \leq t' \leq t$, where t_0 is the time the data element was introduced in the network. To analyse network coverage we extend the model of an individual node in Fig. 10 with four more states. These states are $I0$, $I1$, $I2$ and $I3$. A gossip node is in state Ii if the data element is not in its cache and it has never seen the data element since it was introduced in the network. The latter is the case initially for most nodes, hence the name I: Initial O-state. If a node is in one of the other O-states this means that it does not have the data element in its cache currently, but it was in its cache at an earlier point in time, so the node has seen the data element since it was introduced for the first time.

---

[10]   On the left of each equation the new value of the occupancy measure vector at time step $t+1$, on the right the values of m are intended to be those at time t. For notational simplicity $t+1$ and t have been omitted in the equations below.

The probability functions for the outgoing transitions of the Ii nodes are the same as for their companion O-nodes. Also the probability functions of the incoming transitions, when they come from I-states, are the same. There are no incoming transitions from D-nodes of course, since passing by a D-node would mean that the data element has been in the cache of that node. Similarly, there is no transition from an O-state to an I-state.

Fig. 10. Extended push-pull gossip model of individual gossip node with rounds of length 3 (i.e. $G_{max} = 3$) for the analysis of network coverage. Active states are red, passive ones blue.

The probability functions have to be updated slightly to take the two versions of the O-states into account.

$$\text{d\_loss\_reset } m = (m_{O1} + m_{O2} + m_{O3}) * P(OD|DO) * (\text{noc })+$$
$$(m_{I1} + m_{I2} + m_{I3}) * P(OD|DO) * (\text{noc })+$$
$$(m_{D1} + m_{D2} + m_{D3}) * P(OD|DD) * (\text{noc })$$

$$\text{d\_loss\_step } m = (m_{O0} + m_{I0}) * P(DO|OD) * (\text{noc })+$$
$$m_{D0} * P(DO|DD) * (\text{noc })$$

and their dual probabilities, and

$$\text{o\_getd\_reset } m = (m_{D1} + m_{D2} + m_{D3}) * (P(DO|OD) + P(DD|OD)) * (\text{noc })$$

$$\text{o\_getd\_step } m = m_{D0} * (P(OD|DO) + P(DD|DO)) * (\text{noc }).$$

Similarly to the gossip model for replications we can define a 12×12 matrix $\mathbf{K}$ with indexes i, j in $\{0, \cdots, 11\}$. The definition of the non-zero elements of this matrix can be found in the next section, as well as the additional set of difference equations that can be obtained from it.

8.4   Transition matrix and difference equations for the gossip model in Fig. 10

Numbering the states in the model of Fig. 10 as O0=0, O1=1, O2=2, O3=3, D0=4, D1=5, D2=6, D3=7, I0=8, I1=9, I2=10 and I3=11, the non-empty elements of the $\mathbf{K}$ matrix are given by:

$$
\begin{aligned}
K_{0,3} &= 1.0 - (\mathsf{o\_getd\_reset}\ m) \\
K_{0,7} = K_{8,7} &= \mathsf{o\_getd\_reset}\ m \\
K_{1,0} = K_{2,1} = K_{3,2} &= 1.0 - (\mathsf{o\_getd\_step}\ m) \\
K_{1,4} = K_{2,5} = K_{3,6} &= \mathsf{o\_getd\_step}\ m \\
\\
K_{4,3} &= \mathsf{d\_loss\_reset}\ m \\
K_{4,7} &= 1.0 - \mathsf{d\_loss\_reset}\ m \\
K_{5,0} = K_{6,1} = K_{7,2} &= \mathsf{d\_loss\_step}\ m \\
K_{5,4} = K_{6,5} = K_{7,6} &= 1.0 - \mathsf{d\_loss\_step}\ m \\
\\
K_{9,4} = K_{10,5} = K_{11,6} &= \mathsf{o\_getd\_step}\ m \\
K_{9,8} = K_{10,9} = K_{11,10} &= 1.0 - (\mathsf{o\_getd\_step}\ m)
\end{aligned}
$$

We also obtain the following set of additional four difference equations[10] for the I-states of the model, representing the occupancy measure of state I0 by m_{I0}, I1 by m_{I1}, I2 by m_{I2} and I3 by m_{I3}:

$$
\begin{aligned}
m_{I0} &= m_{I1} - m_{I2} * (\mathsf{o\_getd\_step}\ m) \\
m_{I1} &= m_{I2} - m_{I2} * (\mathsf{o\_getd\_step}\ m) \\
m_{I2} &= m_{I3} - m_{I3} * (\mathsf{o\_getd\_step}\ m) \\
m_{I3} &= m_{I0} - m_{I0} * (\mathsf{o\_getd\_reset}\ m)
\end{aligned}
$$

8.5   Transition matrix and difference equations for the gossip model in Fig. 5 (right)

Numbering the states in the model of Fig. 5 (right) as O=0, D=1, I=2 the non-empty elements of the $\mathbf{K}$ matrix are given by:

$$
\begin{aligned}
K_{0,1} = K_{2,1} &= \mathsf{get}\ m \\
K_{1,0} &= \mathsf{loose}\ m \\
K_{0,0} = K_{2,2} &= 1.0 - (\mathsf{get}\ m) \\
K_{1,1} &= 1.0 - (\mathsf{loose}\ m)
\end{aligned}
$$

Representing the occupancy measure of state O by m_O, D by m_D, I by m_I we obtain the following set of difference equations[10] for the model in Fig. 5 (right):

$$m_O = m_O - m_O * (\text{get } m) + m_D * (\text{loose } m)$$
$$m_D = m_D + m_O * (\text{get } m) - m_D * (\text{loose } m)$$
$$m_I = m_I - m_I * (\text{get } m)$$

8.6 Transition matrix and difference equations for the gossip model in Fig. 7

Numbering the states in the model of Fig. 7 as O=0, D=1, I=2, FD=3, PD=4, LD=5, the non-empty elements of the $\mathbf{K}$ matrix are given by:

$$K_{0,1} = K_{2,1} = K_{3,1} = K_{5,1} = \text{get\_rep } m$$
$$K_{1,0} = \text{loose\_rep } m$$
$$K_{0,0} = K_{2,2} = 1.0 - (\text{get\_rep } m) - (\text{get\_exc } m)$$
$$K_{0,5} = K_{2,3} = \text{get\_exc } m$$

$$K_{1,1} = 1.0 - (\text{loose\_rep } m)$$
$$K_{3,0} = K_{5,0} = \text{loose\_exc } m$$
$$K_{3,3} = K_{5,5} = 1.0 - (\text{loose\_exc } m) - (\text{get\_rep } m)$$
$$K_{4,4} = 1.0$$

Representing the occupancy measure of state O by m_O, D by m_D, I by m_I, FD by m_{FD}, PD by m_{PD} and LD by m_{LD}, we also obtain the following set of difference equations[10] for the model in Fig. 7:

$$
\begin{aligned}
m_O &= m_O - m_O * ((\text{get\_rep } m) + (\text{get\_exc } m)) + m_D * (\text{loose\_rep } m) \\
&\quad + (m_{FD} + m_{LD}) * (\text{loose\_exc } m) \\
m_D &= (m_O + m_I + m_{FD} + m_{LD}) * (\text{get\_rep } m) + m_D - m_D * (\text{loose\_rep } m) \\
m_I &= m_I - m_I * (\text{get\_rep } m + \text{get\_exc } m) \\
m_{FD} &= m_{FD} - m_{FD} * (\text{get\_rep } m) + m_I * (\text{get\_exc } m) - m_{FD} * (\text{loose\_exc } m) \\
m_{PD} &= m_{PD} \\
m_{LD} &= m_{LD} + m_O * (\text{get\_exc } m) - m_{LD} * ((\text{loose\_exc } m) + (\text{get\_rep } m))
\end{aligned}
$$

8.7 Detailed proofs

To prove: $\text{dks}(\mu(t)) + \text{dls}(\mu(t)) = 1$ for all t.. In the following $\text{frc}(X)$ denotes the fraction of nodes in state X.

$$\text{dks}(\mu(t)) + \text{dls}(\mu(t))$$

$=$ {By Defs. of dks and dls}

$$(\text{frc}(O0) + \text{frc}(D0)) * (1 - e^{-2*(\text{frc}(O0) + \text{frc}(D0))}) +$$

$$(1 - (\text{frc}(O0) + \text{frc}(D0))) +$$

$$\text{frc}\,(O0)*(P(OD|OD)+P(DD|OD))*e^{-2*(\text{frc}\,(O0)+\text{frc}\,(D0))}+$$

$$\text{frc}\,(D0)*(P(OD|DD)+P(DD|DD))*e^{-2*(\text{frc}\,(O0)+\text{frc}\,(D0))}+$$

$$\text{frc}\,(O0)*P(DO|OD)*e^{-2*(\text{frc}\,(O0)+\text{frc}\,(D0))}+$$

$$\text{frc}\,(D0)*P(DO|DD)*e^{-2*(\text{frc}\,(O0)+\text{frc}\,(D0))}$$

$= \quad \{\text{Use that } \text{frc}\,(D_i)(t)=\frac{\text{frc}\,(D)(t)}{gmax+1}\text{and } \text{frc}\,(O_i)(t)=\frac{\text{frc}\,(O)(t)}{gmax+1}\}$

$$(\frac{\text{frc}\,(O)(t)}{gmax+1}+\frac{\text{frc}\,(D)(t)}{gmax+1})*(1-*e^{-2/(gmax+1)})+$$

$$(1-(\frac{\text{frc}\,(O)(t)}{gmax+1}+\frac{\text{frc}\,(D)(t)}{gmax+1}))+$$

$$\frac{\text{frc}\,(O)(t)}{gmax+1}*(P(OD|OD)+P(DD|OD))*e^{-2/(gmax+1)}+$$

$$\frac{\text{frc}\,(D)(t)}{gmax+1}*(P(OD|DD)+P(DD|DD))*e^{-2/(gmax+1)}+$$

$$\frac{\text{frc}\,(O)(t)}{gmax+1}*P(DO|OD)*e^{-2/(gmax+1)}+$$

$$\frac{\text{frc}\,(D)(t)}{gmax+1}*P(DO|DD)*e^{-2/(gmax+1)}$$

$= \quad \{\text{Simplify}\}$

$$(\frac{\text{frc}\,(O)(t)}{gmax+1}+\frac{\text{frc}\,(D)(t)}{gmax+1})*(1-e^{-2*(\text{frc}\,(O0)+\frac{\text{frc}\,(D)(t)}{gmax+1})})+$$

$$(1-(\frac{\text{frc}\,(O)(t)}{gmax+1}+\frac{\text{frc}\,(D)(t)}{gmax+1}))+$$

$$\frac{\text{frc}\,(O)(t)}{gmax+1}*(P(OD|OD)+P(DD|OD)+P(DO|OD))*e^{-2/(gmax+1)}+$$

$$\frac{\text{frc}\,(D)(t)}{gmax+1}*(P(OD|DD)+P(DD|DD)+P(DO|DD))*e^{-2/(gmax+1)}$$

$= \quad \{\text{Use } P(OD|OD)+P(DD|OD)+P(DO|OD)=1 \text{ and}$

$\quad\quad P(OD|DD)+P(DD|DD)+P(DO|DD)=1\}$

$$(\frac{\text{frc}\,(O)(t)}{gmax+1}+\frac{\text{frc}\,(D)(t)}{gmax+1})*(1-e^{-2*(\text{frc}\,(O0)+\frac{\text{frc}\,(D)(t)}{gmax+1})})+$$

$$(1-(\frac{\text{frc}\,(O)(t)}{gmax+1}+\frac{\text{frc}\,(D)(t)}{gmax+1}))+$$

$$\frac{\text{frc}\,(O)(t)}{gmax+1}*e^{-2/(gmax+1)}+$$

$$\frac{\text{frc}\,(D)(t)}{gmax+1}*e^{-2/(gmax+1)}+$$

$= \quad \{\text{Simplify}\}$

$$1$$

To prove: $P(OD|OD)+P(DD|OD)+P(DO|OD)=1$

$$P(OD|OD)+P(DD|OD)+P(DO|OD)=1$$

$=$ {Defs. of $P\_a'b'\_ab$}

$\frac{c-s}{c} + \frac{s}{c} * \left(\frac{c-s}{n-s} + \frac{n-c}{n-s}\right)$

$=$ {Simplify}

$\frac{c-s}{c} + \frac{s}{c} * \frac{n-s}{n-s}$

$=$ {Simplify}

$\frac{c-s}{c} + \frac{s}{c}$

$=$ {Simplify}

1

The proof for $P(OD|DD) + P(DD|DD) + P(DO|DD) = 1$ is very similar.