An Analytical Model of Information Dissemination for a Gossip-based Wireless Protocol

Rena Bakhshi, Daniela Gavidia, Wan Fokkink, and Maarten van Steen

Department of Computer Science, Vrije Universiteit Amsterdam, Netherlands
{rbakhshi,daniela,wanf,steen}@few.vu.nl

Abstract. We develop an analytical model of information dissemination for a gossip protocol. With this model we analyse how fast an item is replicated through a network. We also determine the optimal size of the exchange buffer, to obtain fast replication. Our results are confirmed by large-scale simulation experiments.

1 Introduction

Today, large-scale distributed systems consisting of thousands of nodes are commonplace, due to the wide availability of high-performance and low-cost devices. Such systems are highly dynamic in the sense that nodes are continuously in flux, with new nodes joining and existing nodes leaving.

In practice, large-scale systems are often emulated to discover correlations between design parameters and observed behaviour. Such experimental results provide essential data on system behaviour. However, they usually show only behaviour of a particular implementation, and can be time consuming. Moreover, in general experiments do not give a good understanding of the emergent behaviour of the system, and into how parameter settings influence the extra-functional properties of the system. As a result, it is very difficult to predict what the effects of certain design decisions are, as it is practically infeasible to explore the full range of input data. A challenge is to develop analytical models that capture (part of) the behaviour of a system, and then subsequently optimize design parameters following an analytical rather than an experimental approach.

We are interested in developing and validating analytical models for gossip-based systems (cf. [1]). These systems rely on epidemic techniques for the communication and exchange of information. These communication protocols, while having simple specifications, show complex and often unexpected behaviour when executed on a large scale. Our analytical models of gossip protocols need to be realistic, yet, sufficiently abstract to allow for easy prediction of systems behaviour. By 'realistic' we mean that they can be applied to large-scale systems and can capture functional and extra-functional behaviour such as replication, coverage, convergence, and other system dynamics (see [2]). Such models are amenable for mathematical analysis, to make precise predictions. Furthermore,

we will exploit the fact that because an analytical model presents an abstraction of the original protocol, a simulation of the model tends to be much more efficient (in computation time and memory consumption) than a simulation of an implementation of this protocol.

In this paper, we develop an analytical model of a shuffle protocol from [3], which was developed to disseminate data items to a collection of wireless devices, in a decentralized fashion. A decentralized solution considerably decreases the probability of information loss or unavailability that may occur due to a single point of failure, or high latency due to the overload of a node. Nodes executing the protocol periodically contact each other, according to some probability distribution, and exchange data items. The latter is important to achieve the push-pull-based approach, which has a better performance than a pure push or pull approach [4, 5]. Concisely, a node initiates a contact with its random neighbour, pulls a random subset of items from the contacted node, simultaneously pushing its own random subset of items. Replication ensures the availability of the data items even in the face of dynamic behaviour, which is characteristic of wireless environments. And since nodes relocate data in a random fashion, nodes will eventually see all data items.

The central point of our study is a rigorous probabilistic analysis of information dissemination in a wireless network using the aforementioned protocol. The behaviour of the protocol is modelled on an abstract level as pairwise node interactions. When two neighbouring nodes interact with each other (gossip), they may undergo a state transition (exchange items) with a certain probability. The transition probabilities depend on the probability that a given item in a node's cache has been replaced by another item after the shuffle. We calculated accurate values for these probabilities. We also determined a close approximation that is expressed by a much simpler formula, as well as a correction factor for this approximation, allowing for precise error estimations. Thus we obtain a better understanding of the emergent behaviour of the protocol, and how parameter settings influence its extra-functional behaviour.

We investigated two properties characterizing the protocol: the number of nodes that have 'seen' a given item over time (coverage), and the number of replicas of this item in the network at a certain moment in time (replication). Using the values of the transition probabilities, we determined the optimal number of items to exchange per gossip, for a fast convergence of coverage and replication. Moreover, we determined formulas that capture the dissemination of an item in a fully connected network. All our modelling and analysis results are confirmed by large-scale simulations, in which simulations based on our analytical models are compared with running the actual protocol. To the best of our knowledge, we are the first to develop an accurate, realistic formal model that can be used to optimally design and fine-tune a given wireless gossip protocol. In this sense, our main contribution is demonstrating the feasibility of a model-driven approach to developing real-world gossip protocols.

The paper is structured as follows. Sec. 2 explains the shuffle protocol. In Sec. 3 the analytical model is developed and exploited. Sec. 4 discusses the results

of our experimental evaluations. Sec. 5 presents a continuous-time perspective of replication. And Sec. 6 contains the conclusions. Due to space restrictions, several parts of the full version of this paper, available as [6], were omitted: notably an in-depth discussion on replication and coverage, the calculation of the precise formula for the probability of dropping an item, the calculation of the correction factor between this formula and its simplified estimation, and a continuous-time perspective of coverage.

2 A Gossip-based Protocol for Wireless Networks

This section describes the shuffle protocol introduced in [3]. It is a gossip protocol to disseminate data items to a collection of wireless devices. The protocol relies on replication to ensure the availability of data items in the face of dynamic behaviour, which is characteristic of wireless environments.

The system consists of a collection of wireless nodes, each of which contributes a limited amount of storage space (which we will refer to as the node's cache) to store data items. The nodes periodically swap (shuffle) data items from their cache with a randomly chosen neighbour. In this way, nodes gradually discover new items as they are disseminated through the network.

Items can be published by any user of the system, and are propagated through the network. Of each data item, several copies may exist in the network. Replication may occur when a node has available storage space to keep an item it just gossiped to a neighbour.

All nodes have a common agreement on the frequency of gossiping. However, there is no agreement on when to gossip. In terms of storage space, we assume that all nodes have the same cache size c. When shuffling, each node sends a fixed number s of the c items in the cache. The gossip exchange is performed as an atomic procedure, meaning that once a node initiates an exchange with another node, these pair of nodes cannot become involved in another exchange until the current exchange is finished.

In order to execute the protocol, the initiating node needs to contact a gossiping partner. We describe the protocol from the point of view of each participating node. We refer to [3] for a more detailed description.

Node A initiates the shuffle by executing the following steps:

1. picks a neighbouring node B at random;
2. sends s randomly selected items from its local cache to B;
3. receives s items from the local cache of B;
4. checks whether any of the received items are already in its cache; if so, these received items are eliminated;
5. adds the rest of the received items to the local cache; if the total number of items exceeds cache size c, removes items among the ones that were sent by A to B, but not those that were also received by A from B, until the cache contains c items.

In response to being contacted by A, node B consecutively executes steps 3, 2, 4 and 5 above, with all occurrences of A and B interchanged.

According to the protocol, each node agrees to keep the items received from a neighbour. Given the limited storage space available in each node, keeping the items received during an exchange implies discarding some items that the node has in its cache. By picking the items to be discarded from the ones that have been sent to the neighbour, the conservation of data in the network is ensured.

3 An Analytical Model of Information Dissemination

We analyse dissemination of a generic item d in a network in which the nodes execute the shuffling protocol.

3.1 Probabilities of state transitions

We present a model of the shuffle protocol that captures the presence or absence of a generic item d after shuffling of two nodes A and B. There are four possible states of the caches of A and B before the shuffle: both hold d, either A's or B's cache holds d, or neither cache holds d.

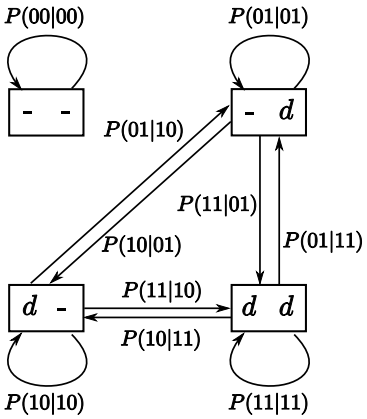


Fig. 1. Symbolic representation for caches of gossiping nodes.

We use the notation $P(a_2b_2|a_1b_1)$ for the probability that from state a_1b_1 after a shuffle we get to state a_2b_2, with $a_i, b_i \in \{0, 1\}$. The indices a_1, a_2 and b_1, b_2 indicate the presence (if equal to 1) or the absence (if equal to 0) of a generic item d in the cache of an initiator A and the contacted node B, respectively. For example, $P(01|10)$ means that node A had d before the shuffle, which then moved to the cache of B, afterwards. Due to the symmetry of information exchange between nodes A and B in the shuffle protocol, $P(a_2b_2|a_1b_1) = P(b_2a_2|b_1a_1)$.

Fig. 1 depicts all possible outcomes for the caches of gossiping nodes as a state transition diagram. If before the exchange A and B do not have d ($a_1b_1 = 00$), then clearly after the exchange A and B still do not have d ($a_2b_2 = 00$). Otherwise, if A or B has d ($a_1 = 1 \vee b_1 = 1$), the shuffle protocol guarantees that after the exchange A or B still has d ($a_2 = 1 \vee b_2 = 1$). Therefore, the state $(-, -)$ has a self-transition, and no other outgoing or incoming transitions.

We determine values for all probabilities $P(a_2b_2|a_1b_1)$. They are expressed in terms of probabilities P_{select} and P_{drop}. Here P_{select} expresses the chance of an item to be selected by a node from its local cache when engaged in an exchange. And P_{drop} represents a probability that an item which can be overwritten (meaning that it is in the exchange buffer of its node, but not of the other node in the shuffle) is indeed overwritten by an item received by its node in the shuffle. Due to the symmetry of the protocol, these probabilities are the same for both

initiating and contacted nodes. In Sec. 3.2, we will calculate P_{select} and P_{drop}. We write $P_{\neg select}$ for $1 - P_{select}$ and $P_{\neg drop}$ for $1 - P_{drop}$.

As explained above, $P(00|00) = 1$. We now focus on the case where $a_1b_1 = 01$, meaning that before shuffling, a copy of d is only in the cache of B.

$a_2b_2 = 01$**:** B did not select (to send) d and, thus, B did not overwrite d; i.e. $P(01|01) = P_{\neg select}$.
$a_2b_2 = 10$**:** B selected d and dropped it; i.e. $P(10|01) = P_{select} \cdot P_{drop}$.
$a_2b_2 = 11$**:** B selected d and kept it; i.e. $P(11|01) = P_{select} \cdot P_{\neg drop}$.
$a_2b_2 = 00$**:** as said, completely discarding d is impossible; i.e. $P(00|01) = 0$.

Due to the symmetry, the case $a_1b_1 = 10$ is similar. We now deal with the case where $a_1b_1 = 11$, meaning that before shuffling, d is in the caches of A and B.

$a_2b_2 = 01$**:** A selected d and dropped it, and B did not select d; i.e. $P(01|11) = P_{select} \cdot P_{drop} \cdot P_{\neg select}$.
$a_2b_2 = 10$**:** symmetric to the previous one: $P(10|11) = P_{\neg select} \cdot P_{select} \cdot P_{drop}$.
$a_2b_2 = 11$**:** after the shuffle both A and B have d, because either:
- A and B did not select d, i.e. $P_{\neg select} \cdot P_{\neg select}$;
- A and B selected d (thus, both kept it), i.e. $P_{select} \cdot P_{select}$;
- A selected d and kept it and B did not select d: $P_{select} \cdot P_{\neg drop} \cdot P_{\neg select}$;
- symmetric case to the previous one: $P_{\neg select} \cdot P_{select} \cdot P_{\neg drop}$.

Thus, $P(11|11) = P_{\neg select} \cdot P_{\neg select} + P_{select} \cdot P_{select} + 2 \cdot P_{select} \cdot P_{\neg select} \cdot P_{\neg drop}$.
$a_2b_2 = 00$**:** as before, $P(00|11) = 0$.

3.2 Probabilities of selecting and dropping an item

The following analysis assumes that all node caches are full (that is, the network is already running for a while). Moreover, we assume a uniform distribution of items over the network; this assumption is supported by experiments in [3, 4].

Consider nodes A and B engaged in a shuffle, and let B receive the exchange buffer S_A from A. Let k be the number of duplicates (see Fig. 2), i.e. the items of an intersection of the node cache C_B and the exchange buffer of its gossiping partner S_A (i.e. $S_A \cap C_B$). Recall that C_A and C_B contain the same number of items for all A and B, and likewise for S_A and S_B; we use c and s for these values. The total number of different items in the network is denoted as n.

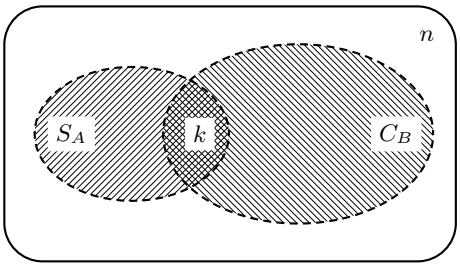


Fig. 2. k items in $S_A \cap C_B$

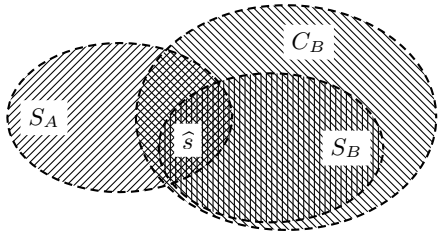


Fig. 3. $\widehat{s}$ items in $S_A \cap S_B$

The probability of selecting an item d in the cache is the probability of a single selection trial (i.e. $\frac{1}{c}$) times the number of selections (i.e. s): $P_{select} = \frac{s}{c}$.

Consider Figs. 2 and 3. The shuffle protocol demands that all items in S_A are kept in C_B after the shuffle. This implies that: a) all items in $S_A \backslash C_B$ will overwrite items in $S_B \subseteq C_B$, and b) all items in $S_A \cap C_B$ are kept in C_B. Thus, the probability that an item from S_B will be overwritten is determined by the probability that an item from S_A is in C_B, but not in S_B. Namely, the items in $S_B \backslash S_A$ provide a space in the cache for items from $S_A \backslash C_B$. We would like to express the probability P_{drop} of a selected item d in $S_B \backslash S_A$ (or $S_A \backslash S_B$) to be overwritten by another item in C_B (or C_A). Due to symmetry, this probability is the same for A and B; therefore, we only calculated the expected probability that an item in $S_B \backslash S_A$ is dropped from C_B. Let $2s \leq c \leq n - s$. Then

$$E[P_{drop}] \quad = \quad \frac{n-c}{\binom{n}{s}} \sum_{k=0}^{s-1} \binom{(n-c)-1}{(s-k)-1} \sum_{\widehat{s}=0}^{k} \frac{\binom{c-s}{k-\widehat{s}}\binom{s}{\widehat{s}}}{s-\widehat{s}} \tag{1}$$

A detailed explanation of how this formula was calculated can be found [6].

3.3 Simplification of P_{drop}

In an effort to simplify the formula for the probability P_{drop} of item in $S_B \backslash S_A$ to be dropped from C_B after a shuffle, we re-examine the relationships between the k duplicates received from a neighbour, the $\widehat{s}$ items of the overlap $S_A \cap S_B$, and P_{drop}. Suppose that $|S_A \cap C_B| = k$, and let's estimate P_{drop} by considering each item from S_A separately, and calculating the probability that the item is a duplicate (i.e., is also in C_B). The probability of an item from S_A to be a duplicate (also present in C_B) is $\frac{c}{n}$. In view of the uniform distribution of items over the network, the items in a node's cache are a random sample from the universe of n data items; so all items in S_A have the same chance to be a duplicate. Thus, the expected number of items in $S_A \cap C_B$ can be estimated by $E[k] = s \cdot \frac{c}{n}$. And the expected number of items in $S_A \cap S_B$ can be estimated by $E[\widehat{s}] = k \cdot \frac{s}{c}$, because only the k items in $S_A \cap C_B$ may end up in $S_A \cap C_B$; $\frac{s}{c}$ captures the probability that an item from C_B is also selected to be in S_B. It follows that the probability of an item in $S_B \backslash S_A$ to be dropped from C_B after the shuffle is $E[P_{drop}] = \frac{s-k}{s-\widehat{s}} = \frac{s - s \cdot \frac{c}{n}}{s - s \cdot \frac{c}{n} \cdot \frac{s}{c}} = \frac{n-c}{n-s}$. This estimate is valid for general $s \leq c \leq n$.

Substituting the expressions for P_{select} and the simplified P_{drop} into the formulas for the transition probabilities in Fig. 1, we obtain:

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

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

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

In order to verify the accuracy of the proposed simplification for $E[P_{drop}]$, we compare the simplification and the accurate formula (1) for different values of

n. We plot the difference of the accurate P_{drop} and the simplification, for cache sizes $c = 250$ and $c = 500$ (Fig. 4).

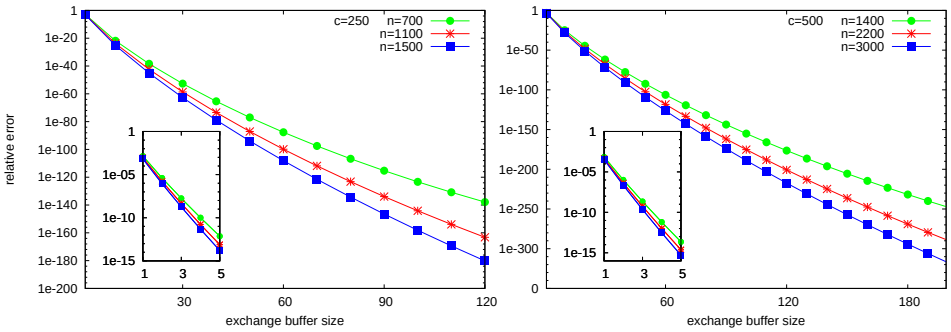


Fig. 4. The difference of the accurate P_{drop} and its approximation, for different values of n and c.

We also investigated how closely the simplified formula $E[P_{drop}] = \frac{n-c}{n-s}$ approximates formula (1). We determined that $E[P_{drop}] = \frac{n-c}{(n-s)+\frac{1}{\gamma}}$, where

$$\gamma = \sum_{d=0}^{s-1} \frac{\binom{n}{d}}{s \cdot \binom{s-1}{d}} = \frac{\binom{n}{s}}{(n-s)+1} \cdot \sum_{d=0}^{s-1} \frac{1}{\binom{n-d}{(s-1)-d}} \tag{2}$$

In [6] it is explained how we discovered γ. Extensive experiments with Mathematica and Matlab indicate that $\frac{n-c}{(n-s)+\frac{1}{\gamma}}$ and formula (1) coincide.

3.4 Optimal size for the exchange buffer

We study what is the optimal value for fast convergence of replication and coverage with respect to an item d. Since d is introduced at only one node in the network, one needs to optimize the chance that an item is duplicated. That is, the probabilities $P(11|01)$ and $P(11|10)$ should be optimized (then $P(01|11)$ and $P(10|11)$ are optimized as well, intuitively because for each duplicated item in a shuffle, another item must be dropped). These probabilities both equal $\frac{s}{c}\frac{c-s}{n-s}$; we compute when the s-derivative of this formula is zero. This yields the equation $s^2 - 2ns + nc = 0$; taking into the account that $s \leq n$, the only solution of

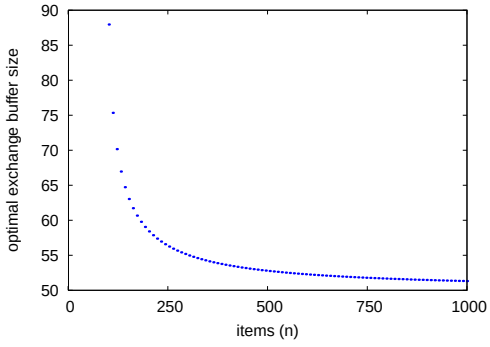


Fig. 5. Optimal value of exchange buffer size, depending on n.

this equation is $s = n - \sqrt{n(n-c)}$. We conclude that this is the optimal value for s to obtain fast convergence of replication and coverage. This will also be confirmed by the experiments and analyses in the following sections.

4 Experimental Evaluation

In order to test the validity of the analytical model of information spread under the shuffle protocol presented in the previous section, we followed an experimental approach. We compared properties observed while running the shuffle protocol in a large-scale deployment with simulations of the model under the same conditions. These experiments show that the analytical model indeed captures information spread of the shuffle protocol. We note that a simulation of the analytical model is much more efficient (in computation time and memory consumption) than a simulation of the implementation of the shuffle protocol.

The experiments simulate the case where a new item d is introduced at one node in a network, in which all caches are full and uniformly populated by $n = 500$ items. They were performed on a network of $N = 2500$ nodes, arranged in a square grid topology (50×50), where each node can communicate only with its four immediate neighbors (to the North, South, East and West). Each node has a cache size of $c = 100$, and sends s items when gossiping. In each round, every node randomly selects one of its neighbors, and updates its state according to the transition probabilities introduced before (Fig. 1). This mimics (the probabilities of) an actual exchange of items between a pair of nodes according to the shuffle protocol. In the experiments, after each gossip round, we measured the total number of occurrences of d in the network (replication), and how many nodes in total have seen d (coverage).

To fill the caches of the nodes with a random selection of items, measurements are initiated after 1000 gossip rounds. In other words, 500 different items are inserted at the beginning of the simulation, and shuffled for 1000 rounds. During this time, items are replicated and the replicas fill the caches of all nodes. At round 1000, a copy of the fresh item d is inserted at a random location, and its spread through the network is tracked over the next 2000 rounds.

Fig. 6 shows the behaviour of both the shuffle protocol and the analytical model in terms of replication and coverage of d, for various values of s. Each curve in the graphs represents the average and standard deviation calculated over 30 runs. The experiments with the model calculate P_{drop} using the simplified formula $\frac{n-c}{n-s}$ described in Sec. 3.3. Clearly the results obtained from the model (right) resemble closely the ones from executing the protocol (left).

In all cases, the network converges to a situation in which there are 500 copies of d, meaning that replication is $\frac{500}{2500} = 0.2$; this agrees with the fact that $\frac{c}{n} = \frac{100}{500} = 0.2$. Moreover, replication and coverage display the fastest convergence when $s = 50$; this agrees with the fact that $n - \sqrt{n(n-c)} = 500 - \sqrt{500 \cdot 400} \approx 50$ (cf. Sec. 3.4).

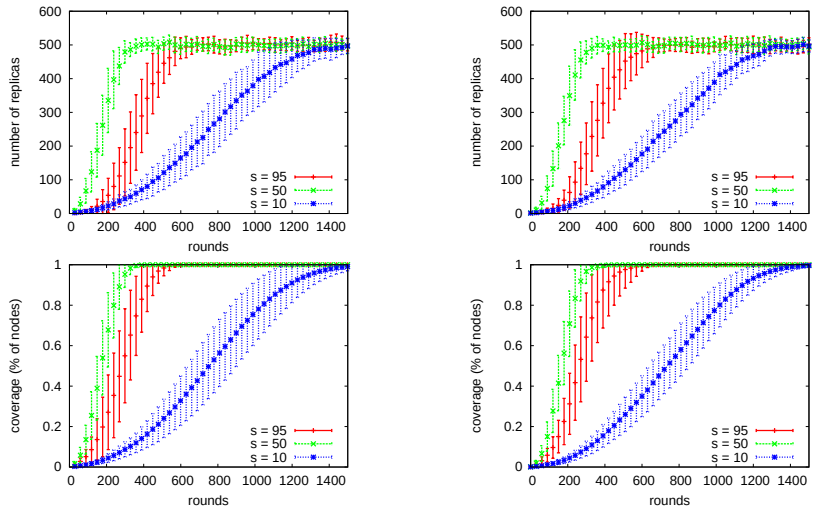


Fig. 6. The shuffle protocol (left) and the model (right), for $N = 2500$, $n = 500$, $c = 100$ and different values of s.

5 Continuous-time Modelling of Replication

In this section we exploit the analytical model of information dissemination to perform a mathematical analysis of replication with regard to the shuffle protocol. For the particular case of a network with full connectivity, where a node can gossip with any other node in the network, we can find explicit expressions for the dissemination of a generic item d in terms of the probabilities presented in Sec. 3. We construct a differential equation that captures replication of item d from a continuous-time perspective. Thus we can determine the long-term behavior of the system as a function of the parameters. In the full version [6], also a differential equation for coverage is determined and exploited.

One node introduces a new item d into the network at time $t = 0$, by placing it into its cache. From that moment on, d is replicated as a consequence of gossiping among nodes. Let $x(t)$ represent the percentage of nodes in the network that have d in their cache at time t, where each gossip round takes one time unit. The variation in x per time unit $\frac{dx}{dt}$ can be derived based on the probability that d will replicate or disappear after an exchange between two nodes, where at least one of the nodes has d in its cache:

$$\frac{dx}{dt} = [P(11|10) + P(11|01)] \cdot (1 - x) \cdot x - [P(10|11) + P(01|11)] \cdot x \cdot x$$

The first term represents duplication of d when a node that has d in its cache initiates the shuffle, and contacts a node that does not have d. The second term represents the opposite situation, when a node that does not have d initiates a shuffle with a node that has d. The third and fourth term in the equation

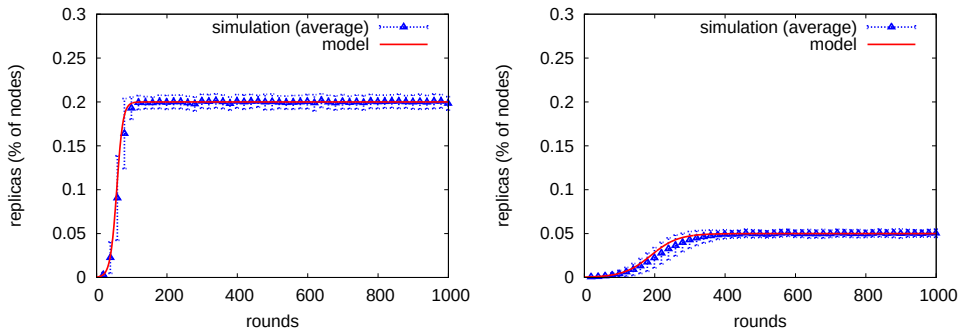
represent the cases where both nodes have d in their cache, and after the exchange only one copy of d remains. Substituting $P(11|10) = P(11|01) = \frac{s}{c}\frac{c-s}{n-s}$ and $P(10|11) = P(01|11) = \frac{s}{c}\frac{n-c}{n-s}\frac{c-s}{c}$, we obtain

$$\frac{dx}{dt} = 2 \cdot \frac{s}{c} \cdot \frac{c-s}{n-s} \cdot x \cdot (1 - \frac{n}{c} \cdot x) \tag{3}$$

The solution of this equation, taking into account that $x(0) = \frac{1}{N}$ with N the number of nodes in the network, is

$$x(t) = \frac{e^{\alpha t}}{(N - \frac{n}{c}) + \frac{n}{c}e^{\alpha t}} \tag{4}$$

where α denotes $2\frac{s}{c}\frac{c-s}{n-s}$. By imposing stationarity, i.e. $\frac{dx}{dt} = 0$, we find the stationary solution $\frac{c}{n}$. This agrees with the fact that the protocol achieves a uniform distribution of items over the network. Namely, since there are Nc cache entries in the network in total, the average number of copies of an individual item in the network converges to $\frac{Nc}{n}$; so replication converges to $\frac{c}{n}$.



500 items 2000 items

Fig. 7. Percentage of nodes in the network with a replica of item d in their cache, for $N = 2500$, $c = 100$, $s = 50$, and $n = 500$ or $n = 2000$.

We evaluate the accuracy of $x(t)$ as a representation of the fraction of nodes carrying a replica of d, by running a series of experiments where $N = 2500$ nodes execute the shuffle protocol, and their caches are monitored for the presence of d. Unlike the experiments in Sec. 4, we assume full connectivity; that is, for each node, all other nodes are within reach. After 1000 rounds, where items are disseminated and replicated, a new item d is inserted at a random node, at time $t = 0$. We track the number of replicas of d for the next 1000 rounds. The experiment is repeated 30 times and the results are averaged. The simulation results and $x(t)$, presented in Fig. 7, show the same initial increase in replicas after d has been inserted, and in both cases the steady state reaches precisely the expected value $\frac{c}{n}$ predicted from the stationary solution.

We repeat the calculation from Sec. 3.4, but now against $x(t)$, to determine which size of the exchange buffer yields the fastest convergence to the

steady-state for both replication and coverage. That is, we search for the s that maximizes the value of $x(t)$. We first compute the derivative of $x(t)$ with respect to s ($z(t,s)$), and then derive the value of s that maximizes $x(t)$, by taking $z(\cdot,m) = \frac{\partial x}{\partial s}|_m = 0$: $z(t,s) = \frac{\partial x}{\partial s} = \frac{2e^{kt}(cN-n)(cn+s(-2n+s))t}{(cN+(-1+e^{kt})n)^2(n-s)^2}$, where $k = 2\frac{s}{c}\frac{c-s}{n-s}$. Let $z(t,s) = 0$. For $t > 0$, $cn = s(2n-s)$. Taking into the account that $s \leq n$, the only solution for this equation is $s = n - \sqrt{n(n-c)}$. So this coincides with the optimal exchange buffer size found in Sec. 3.4.

6 Conclusions

We have demonstrated that it is possible to model a gossip protocol through a rigorous probabilistic analysis of the state transitions of a pair of gossiping nodes. We have shown, through an extensive simulation study, that the dissemination of a data item can be faithfully reproduced by the model. Having an accurate model of node interactions, we have been able to carry out the following:

- After finding precise expressions for the probabilities involved in the model, we provide a simplified version of the transition probabilities. These simplified, yet accurate, expressions can be easily computed, allowing us to simulate the dissemination of an item without the complexity of executing the actual shuffle protocol. These simulations use very little state (only some parameters and variables, as opposed to maintaining a cache) and can be executed in a fraction of the time required to run the protocol.
- The model reveals relationships between system parameters. Armed with this knowledge, we successfully optimize one of the parameters (the size of the exchange buffer) to obtain fast convergence of replication.
- Under the assumption of full connectivity, we are able to use the transition probabilities to model replication and coverage for a generic item. Each property is ultimately expressed as a formula which is shown to display the same behavior as the average behavior of the protocol, verifying the validity of the model.

While gossip protocols are easy to understand, even for a simple push/pull protocol, the interactions between nodes are unexpectedly complex. Understanding these interactions provides insight into the mechanics behind the emergent behavior of gossip protocols. We believe that understanding the mechanics of gossiping is the key to optimizing (and even shaping) the emergent properties that make gossiping appealing as communication paradigm for distributed systems.

References

1. Bakhshi, R., Bonnet, F., Fokkink, W., Haverkort, B.: Formal analysis techniques for gossiping protocols. ACM SIGOPS Oper. Syst. Rev. **41**(5) (2007) 28–36
2. Eugster, P., Guerraoui, R., Kermarrec, A.M., Massoulié, L.: Epidemic Information Dissemination in Distributed Systems. IEEE Computer **37**(5) (2004) 60–67

3. Gavidia, D., Voulgaris, S., van Steen, M.: A Gossip-based Distributed News Service for Wireless Mesh Networks. In: Proc. 3rd IEEE Conf. on Wireless On-demand Network Systems and Services (WONS'06), IEEE (2006) 59–67
4. Jelasity, M., Voulgaris, S., Guerraoui, R., Kermarrec, A.M., van Steen, M.: Gossip-based peer sampling. ACM Trans. Comput. Syst. **25**(3) (2007) 8
5. Karp, R., Schindelhauer, C., Shenker, S., Vocking, B.: Randomized rumor spreading. In: Proc. 41st Symp. on Found. of Comput. Sci. (FOCS'00), IEEE (2000) 565–574
6. Bakhshi, R., Gavidia, D., Fokkink, W., van Steen, M.: An analytical model of information dissemination for a gossip-based wireless protocol `http://www.few.vu.nl/~rbakhshi/gfull.pdf`.