Correctness of Gossip-Based Membership under Message Loss

Maxim Gurevich*
Dept. of Electrical Engineering
Technion, Haifa 32000, Israel
gmax@tx.technion.ac.il

Idit Keidar
Dept. of Electrical Engineering
Technion, Haifa 32000, Israel
idish@ee.technion.ac.il

ABSTRACT

Due to their simplicity and effectiveness, gossip-based membership protocols have become the method of choice for maintaining partial membership in large P2P systems. A variety of gossip-based membership protocols were proposed. Some were shown to be effective empirically, lacking analytic understanding of their properties. Others were analyzed under simplifying assumptions, such as lossless and delay-less network. It is not clear whether the analysis results hold in dynamic networks where both nodes and network links can fail.

In this paper we try to bridge this gap. We first enumerate the desirable properties of a gossip-based membership protocol, such as view uniformity, independence, and load balance. We then propose a simple *Send & Forget* protocol, and show that even in the presence of message loss, it achieves the desirable properties.

Categories and Subject Descriptors: H.3.4: Distributed systems.

General Terms: Algorithms, Reliability, Theory.

Keywords: Peer-to-peer, membership, gossip, random sampling.

1. INTRODUCTION

Large-scale dynamic systems are nowadays being deployed in many places, including peer-to-peer networks over the Internet, in data centers, and computation grids. Such systems are subject to *churn*, i.e., their membership constantly changes, as nodes dynamically join and leave. Moreover, such systems are often comprised of unreliable components, where node failures and message losses are frequent.

In order to allow nodes to communicate with each other, each node must know the ids (for example, IP addresses and ports), of some other nodes. Such ids are stored at each node in a *local view* (sometimes called membership), or *view* for short. In large systems, it is uncommon to store full views including all nodes in the system, not only because of the amount of memory this would require, but also because of the high maintenance overhead that churn would induce. Instead, one typically stores small views, e.g., logarithmic in system size [8, 2]. Local views are maintained by a distributed *group membership protocol*.

The views of all nodes induce a *membership graph* (overlay network), over which communication takes place. Two nodes are *neighbors* if one of their views includes the id of the other. The properties of local views have significant consequences for the respective graph's diameter, connectivity, load-balance, and robustness. Our goal in this paper is to mathematically analyze the proprieties of such views, and in particular, to understand the impact that message loss has on these properties.

We begin, in Section 2, by identifying the goals that a membership service strives to achieve: First, to bound the load on each node, each node has to maintain a *small view* and have a *bounded degree* (number of neighbors). Additionally, the "holy grail" for a membership service is to choose view entries independently of each other (we call this *spatial independence*) and uniformly at random [8, 21, 7]. Indeed, such choices result in an expander graph, with good connectivity and robustness, and low diameter [9], ensuring fast and reliable communication. Note that in a dynamic system subject to churn, local views must evolve to reflect joining nodes and exclude ones that left or failed, and the system should converge to independent uniform views from *any* initial topology.

Beyond maintaining the membership graph for communication, independent random node id samples are useful for a variety of additional applications, such as gathering statistics, gossip-based aggregation, and choosing locations for data caching [17, 12, 5]. Such applications constantly require fresh random node ids, independent of past views, which requires views to evolve even in the absence of churn or failures. We thus identify an additional goal for a membership service: *temporal independence*– evolving into new graphs whose dependence on the past decays rapidly.

The most common approach to maintaining small local views is using *gossip-based membership* protocols [11, 8, 2, 23, 16]. In such protocols, nodes exchange ("gossip about") ids from their views with their neighbors, and use this information to update their views (see Section 3). Such protocols make random choices, and their evolution is therefore a random process. Gossip-based membership has been empirically shown to lead to good load balance of node degrees [8,

[*] Supported by the Eshkol Fellowship of the Israeli Ministry of Science.

PODC'09, August 10–12, 2009, Calgary, Alberta, Canada.
Copyright 2009 ACM 978-1-60558-396-9/09/08 ...$10.00.

16], and certain variants of gossip were proven to ensure low probability for partitions [2]. On the other hand, most gossip-based protocols do, in fact, induce spatial dependencies among neighboring nodes. This is because an id that is gossiped to a neighbor typically remains in the sender's view.

Spatial dependencies can be eliminated by deleting ids sent to a neighbor. In order to avoid having unused entries in views, this is usually done in actions involving bidirectional communication, where the id received in a reply replaces the sent id [2, 18, 19]. However, such actions were previously analyzed under the assumption that they occur *atomically*, without overlapping in time with any other action, even though they involve multiple nodes. In practice, it is unclear how overlap can be avoided, as protocol actions are initiated from different nodes asynchronously, and a node might receive a message initiating a new action while it is already engaged in another. Moreover, implementing such atomic actions requires bookkeeping at each node, and is of course impossible in the presence of message loss [14] or node failures.

In Section 4, we present a model for studying gossip-based membership without atomicity assumptions. We follow [18, 19], and model protocol actions as random graph transformations. In order to apply this methodology to real systems, we break up protocol actions into steps that can be executed atomically at a single node, allowing the analysis to account for message loss.

In Section 5, we present *Send & Forget* (*S&F*), a simple and practical protocol that eliminates bidirectional communication, at the cost of allowing for unused (empty) entries in views. Message loss increases the number of unused entries. The protocol compensates for loss by creating new, dependent view entries. The goal is to create as little dependencies as possible.

In Section 6, we analyze node degree distributions induced by *S&F*. Our analysis shows that *S&F* can operate with small views– constant (e.g., with 40 entries), or logarithmic in system size. It further shows that the distribution of node degrees is very well balanced– close to the binomial distribution.

In Section 7 we study the distribution of membership graphs the protocol evolves to (i.e., the protocol's properties in the steady state). We define a Markov Chain (MC) on the global states (membership graphs) reachable by *S&F* starting from any weakly connected membership graph. We show that without loss, *S&F* achieves the desired properties of uniformity and independence. With positive loss, uniformity still holds but there exist spatial dependencies among entries in the same view as well as among views of neighboring nodes. These dependencies increase very moderately with the loss rate: The fraction of dependent entries in views is bounded, and grows like twice the loss rate. As the loss is typically in the order of 1% [22, 4], the vast majority of view entries are expected to be independent. From this bounded spatial dependence, we prove that the temporal independence is preserved. We show that in a system of size n, starting from a random state (membership graph) G in the MC, once each node initiates $O(s \log n)$ actions, where s is a view size and n is the number of nodes in the system, the system evolves to a state whose dependence on G can be made arbitrarily small. For space limitations, some formal proofs are deferred to the full paper [15].

In summary, we make the following contributions:

- We spell out the desired properties of membership protocols that maintain small views.
- We provide a model for studying membership graph evolution with non-atomic protocol actions.
- We present a practical membership protocol, *S&F*, which is amenable to formal analysis.
- In the absence of message loss, *S&F* provides all the desired properties of a membership service.
- We present the first formal analysis of a membership protocol in the presence of message loss. The salient properties of *S&F* are preserved even under reasonable loss rates.

2. GOALS FOR A DISTRIBUTED MEMBERSHIP SERVICE

We consider a dynamic distributed system with up to n nodes active at any given time. When using a distributed membership service, no single participant has the complete membership information. Instead, each node u maintains a local view – a multiset, $u.\mathit{lv}$, of s node ids, also denoted $u.\mathit{lv}[1..s]$. We say that u is an *in-neighbor* of v, and that v is an *out-neighbor* of u, if $v \in u.\mathit{lv}$. We denote such a view entry by (u, v). We say that two nodes are *neighbors* if one of them is either an in- or out-neighbor of another. The *outdegree* of u, denoted $d(u)$, is the number of out-neighbors u has. Since some view entries might be empty, this number may be smaller than s. Similarly, u's *indegree*, denoted $d_{\text{in}}(u)$, is the number of in-neighbors u has.

We now formalize the desirable properties of a distributed membership service. First, in large systems it is infeasible (in terms of memory, bandwidth, and processing time) for each node to maintain the full membership information. We thus require:

Property (M1 - Small Views)**.** *The view size $s \ll n$.*

Typically, logarithmic size views are used in order to ensure fast dissemination of gossiped information [8]. Other applications work with constant-size views [21]. Property M1 has to hold at all times.

We next define the load-balance, uniformity, and independence properties of the membership graph. Note that nodes can be expected to be uniformly and independently represented in views only after they have been in the system "long enough" for their representation to spread in the system; these properties cannot be expected to hold for newly joined or recently departed nodes whose ids are still included in views. Therefore, similarly to previous studies [6], we require the following properties to hold only if churn ceases from some point onward. For simplicity, we model this by considering a static system of n nodes $u_1, u_2, \ldots, u_n$. Note that our load-balance, uniformity, and spatial independence properties are required to eventually hold, starting from *any* initial state, and thus we effectively deal with churn that affects the initial topology.

The number of messages received by a node (sent by the membership protocol or by an application) is proportional to the number of its in-neighbors. We therefore require load balancing of indegrees:

Property (M2 - Load Balance)**.** *Starting from any initial state, eventually, the variance of node indegrees is bounded.*

The main quality measure of a local view is how well it approximates an independent and identically distributed (IID) uniform sample of the nodes The next two properties stipulate that views should converge to IID uniform ones, from any state.

Property (M3 - Uniform Sample)**.** *Starting from any initial state, eventually, for each u, v, w,*

$$\Pr(v \in u.\mathit{lv}) \quad = \quad \Pr(w \in u.\mathit{lv}).$$

Note the difference between M2 and M3: M2 means that eventually, in *each* membership graph each node is represented near-uniformly in other nodes' views. M3, on the other hand, implies that after the system runs for a long time, every id eventually has the same likelihood of appearing in any given view entry.

Uniformity, by itself, does not imply independence among view entries of the same node or of different nodes at the same time. Since typical membership protocols exchange data between neighbors, the most likely dependencies are within the same view, or among the views of neighboring nodes. We say that two nonempty view entries $u.\mathit{lv}[i]$ and $v.\mathit{lv}[j]$ are *independent* of each other if

$$\Pr(u.\mathit{lv}[i] = w | v.\mathit{lv}[j] = w) \quad = \quad \Pr(u.\mathit{lv}[i] = w).$$

By slight abuse of terminology, we simply label edges in a membership graph as dependent without specifying what edges they depend on, as follows: (1) All self-edges ($u.\mathit{lv}[i] = u$) are *dependent*; (2) For $v = u$ or $v \in u.\mathit{lv}$, if $u.\mathit{lv}[i]$ is not independent of $v.\mathit{lv}[j]$ for some j then we say that one of $u.\mathit{lv}[i]$ or $v.\mathit{lv}[j]$ is *dependent*. In case of dependencies among several edges, all but one of these edges are considered dependent. Every edge that is not dependent is *independent*. We are now ready to define spatial independence.

Property (M4 - Spatial Independence)**.** *Starting from any initial state, eventually, for each u and $1 \leq i \leq s$ such that $u.\mathit{lv}[i]$ is nonempty, we wish to bound the probability that $u.\mathit{lv}[i]$ is independent.*

Typical membership protocols update only a part of the view in each step. Thus, there is a *temporal* dependence between the views before and after the update. We are interested in protocols that lead to fast dependence decay:

Property (M5 - Temporal Independence)**.** *Starting from an expected initial state (formally defined in Section 4), we wish to bound the number of actions the protocol needs to take in order to reach a state that is independent of the initial state.*

Note that the above bound is weaker than a bound on mixing time, which considers convergence time from an *arbitrary* state, rather than a random one.

3. BACKGROUND: MEMBERSHIP PROTOCOLS

We provide a brief taxonomy of the basic actions of gossip-based membership protocols.

Action initiator. A node u can contact one of its out-neighbors v to either *push* some node id to it, or to *pull* an id from it. The pushed id is added to v's view. In a pull, v is expected to return some id, which u adds to its view. In some protocols, push and pull are combined into a single protocol action [2, 18, 19].

The ids sent. Allavena *et al.* [2] identified two crucial components for a good membership protocol: In a *reinforcement* component, a node adds its own id to an other node's view. Reinforcement leads to a uniform representation of nodes in other nodes' views, and fixes any non-uniformity that might have been caused by a bad initial views or churn. In a *mixing* component, a node adds to its view an id from an other node's view. This component spreads membership information among nodes, thus providing independence.

Note that each of the components can be implemented by either push or pull. While many protocols implement reinforcement by push and mixing by pull, e.g., [2, 19], Lpbcast [8] uses push for both. We do the same in this paper. A practical optimization, made in many protocols, e.g., [8, 2], is performing several actions at once, thus reducing message overhead. Such protocols, however, are difficult to analyze, so most analyses assume that actions are executed serially [2, 18, 19], as we do in this paper.

Protocols also differ in whether the sender deletes the ids it sent from its local view or keeps them. Most protocols, e.g., [8, 2] keep the sent ids, thus inducing dependence between neighbor views. Those that delete the sent ids, e.g., shuffle [1, 19], and flipper [18], are unable to withstand message loss or node failures since the system gradually loses more and more ids. Jelasity *et al.* [16] combine shuffle, which does not create dependencies but may lose ids, with regular push-pull, which creates dependencies but is immune to loss. In their approach, shuffle operations constitute a predetermined fraction of all operations, regardless of actual loss or churn. In contrast, in *S&F*, dependencies are created only to compensate for actually lost ids, and can be kept arbitrarily low with no loss.

Other sampling approaches. An important advantage of gossip-based membership is the use of local operations, where each node communicates only with its immediate neighbors. An alternative (non-local) approach is to use random walks (RWs) (on the membership graph) to obtain new ids for local views [13, 5, 20]. However, a RW requires many steps, and its correctness depends on the graph topology; if the actual topology is different from the assumed one, then the sample may be far from uniform [13]. Moreover, the analysis of RW convergence ignores the dynamic nature of the graph; recent work suggests that RWs may be much less effective on dynamic graphs [3]. In this paper, we consider local operations only.

Another characteristic of gossip-based membership protocols is that they use the local view for two purposes: (1) to provide node id samples to the application, and (2) to define the communication graph over which messages of the gossip protocol itself are transmitted. It is possible to separate the two. For example, Brahms [6] uses fast evolving local views, which might be non-uniform, and complements them with membership samples, which converge to uniform ones over time. However, the latter do not provide temporal independence, as they are designed to persist rather than evolve. We note that Brahms was designed for Byzantine settings, where maintaining uniform views is challenging. In this paper, we consider benign settings, and are interested in evolving yet uniform local views.

4. MODELING MEMBERSHIP PROTOCOLS BY GRAPH TRANSFORMATIONS

We model membership as a directed multigraph $G = (V, E)$ where vertices represent nodes and edges represent membership information: E is a multiset containing an edge (u, v) for each u and v such that $v \in u.\mathit{lv}$, with the multiplicity equal to the multiplicity of v in $u.\mathit{lv}$. Unless specified otherwise, we assume the graph to be weakly connected. That is, there is an undirected path between every two nodes.

Protocol actions can be described as transformations on graph G. For example, a push action of w's id from u to v adds an edge (v, w), and pulling id w by u from v adds an edge (u, w).

We consider only memoryless random transformations. That is, each transformation allowed by a particular protocol occurs with a probability that depends only on the current membership graph. Every protocol thus defines a Markov Chain (MC) $\tilde{G}(0), \tilde{G}(1), \ldots$, where $\tilde{G}(i)$ represents the distribution of the membership graphs after the i-th action of the protocol. We analyze a protocol's MC graph, where vertices are all possible membership graphs, and edge weights are transition probabilities of the protocol. A stationary distribution π of such an MC (assuming it exists) describes the steady state of the system. We thus can analyze the properties of an expected (according to π) membership graph and and the extent to which it satisfies the desired properties defined in Section 2.

4.1 Distributed Operations

Because each node's knowledge of the system is partial, only a limited set of transformations can occur as a result of a distributed protocol in any given state. Protocol actions are composed of steps, as defined below:

Protocol steps. A *step* is a transformation that can be implemented at a single node and consists of the following three elements: (1) receiving of 0 or 1 messages, (2) modifying the local view by adding ids received in the message (including the sender's id) and deleting and duplicating arbitrary ids, and (3) sending 0 or more messages that can include ids received in the message in (1), ids from the current view or from the previous view before performing (2). A key property of a step is that it can be executed atomically, even in an environment with message loss.

Protocol actions. A number of steps can be combined into a protocol *action*, starting with a step of an *initiating* node u, followed by a sequence of steps that receive messages sent in the previous steps. For example, in a push action from u to its out-neighbor v, u's send to v is a step and v's receive and view modification is another step.

Previous analyses, e.g., [2, 18, 19], assumed atomic actions, with no overlap in time. However, guaranteeing atomicity of multi-step actions in a real system may be complex, and is in some cases impossible, e.g., in the presence of message loss or of unreliable nodes and asynchronous communication [14, 10].

Modeling Loss with Non-atomic Actions. We assume there is some probability ℓ that a sent message is not delivered at its destination. We further assume this probability to be unknown to the protocol, identical for all messages, and independent of other messages. We assume that the sender cannot detect that the message it sent was lost, so it cannot retransmit the message. This means that in a multi-step action, each step is executed with probability $1 - \ell$, given that the previous step was executed (except for the first step, which is executed with probability 1).

5. SEND & FORGET PROTOCOL

We present *S&F*, a simple and practical protocol that overcomes loss. *S&F* avoids bidirectional communication within the same action; after it sends a message, it "forgets" about it. Thus, actions at each node are trivially non-overlapping. The protocol running at each node is shown in Figure 1 (u.a.r. stands for uniformly at random). Each node u maintains a view $u.\mathit{lv}$ – an array of size s, where s is even. In order to overcome loss (non-atomic actions), the protocol is parametrized by a threshold d_{L} that sets a lower bound on node outdegree.

The protocol at node u works as follows: the node selects two different entries i and j in its view uniformly at random. If any of them is empty, nothing happens and the views of all the nodes remain unchanged. If both $v = u.\mathit{lv}[i]$ and $w = u.\mathit{lv}[j]$ are nonempty, then u performs the following steps: (1) sends to v a message including its own id and w; and (2) clears both entries i and j in its view, unless $d(u) \le d_{\mathrm{L}}$, in which case we say the entries are *duplicated*. On receiving a message, a node adds both received ids to empty entries in its view, unless $d(u) = s$, in which case we say the received ids are *deleted*. Figure 2 (a)-(b) shows the graph transformation performed by the protocol when sender's and receiver's outdegrees are between d_{L} and s, (which happens most of the times). Figure 2 (c) shows the effect of duplication at the sender; and Figure 2 (d) illustrates message loss or deletion at the receiver.

```
function $S\&F$-InitiateAction$_u()$
select $1 \le i \ne j \le s$ u.a.r.
$v \leftarrow u.\mathit{lv}[i]$
$w \leftarrow u.\mathit{lv}[j]$
if $v \ne \bot$ AND $w \ne \bot$ then
    send $[u, w]$ to $v$
    if $d(u) > d_{\mathrm{L}}$ then
        $u.\mathit{lv}[i] \leftarrow \bot$
        $u.\mathit{lv}[j] \leftarrow \bot$
```

```
function $S\&F$-Receive$_u(v_1, v_2)$
if $d(u) < s$ then
    select $i$ u.a.r. so that $u.\mathit{lv}[i] = \bot$
    select $j$ u.a.r. so that $u.\mathit{lv}[j] = \bot$
    $u.\mathit{lv}[i] \leftarrow v_1$
    $u.\mathit{lv}[j] \leftarrow v_2$
```

Figure 1: The Send *&* Forget protocol at node u.

The purpose of the duplications, controlled by the threshold d_{L}, is to compensate for loss. In the absence of loss, d_{L} can be set to zero, disabling duplications. Under positive loss and without duplications, node outdegrees would gradually decrease, until eventually all nodes become isolated. To prevent such a scenario, the protocol performs duplications and creates new edges in the membership graph instead of lost ones. One might wonder why not fill up empty view entries by replicating ids in the view. We avoid such replications since it increases dependencies among ids in the same

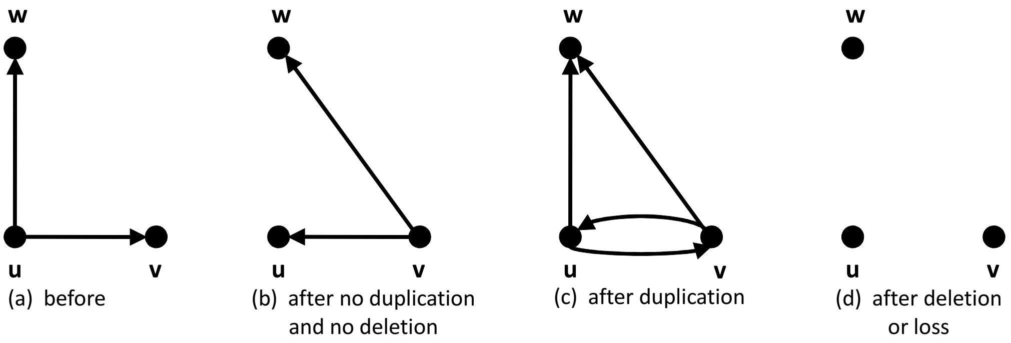

w
u
v
(a) before
w
u
v
(b) after no duplication and no deletion
w
u
v
(c) after duplication
w
u
v
(d) after deletion or loss

Figure 2: Possible outcomes of a transformation of *S&F*, initiated by u sending message $[u, w]$ to v. (a) Before the transformation. Possible states after the transformation where: (b) $d(u) > d_{\mathrm{L}}, d(v) < s$, message delivered; (c) $d(u) = d_{\mathrm{L}}, d(v) < s$, message delivered; (d) $d(u) > d_{\mathrm{L}}$, and $d(v) = s$ or message lost.

view. Instead, we allow the sent ids to remain in the sender's view. Although such duplication still creates dependencies among neighbors' views, it does not directly create redundant parallel edges. As the protocol occasionally creates too many edges, it may need to delete some, when there are no empty view entries to store the received ids. In Section 6, we analyze the impact of d_{L} and s (recall that the view size is bounded by s), which in turn provides a "rule-of-thumb" for selecting their values.

In our analysis, we assume that a central entity repeatedly selects a random node, invokes its *S&F*-InitiateAction$_u$() method, and waits for the completion of *S&F*-Receive$_u(v_1, v_2)$ by the receiving node (in case a message was sent). In practice, a similar behavior can be implemented by each node periodically invoking its *S&F*-InitiateAction$_u$() method when the invocation rate is the same for all nodes. The next proposition follows immediately.

Proposition 5.1. *The probability for every node u, and every two entries in u's view to be chosen in an action is the same.*

6. NODE DEGREE ANALYSIS AND SETTING DEGREE THRESHOLDS

In this section we show that *S&F* satisfies the properties M1 - Small Views (i.e., $s \ll n$) and M2 - Load Balance, defined in Section 2.

We start, in Section 6.1, with assuming that the protocol actions are atomic (no loss), that the views are initialized so that for all u, $d(u) + 2\,d_{\mathrm{in}}(u)$ is constant, and that no edge duplications or deletions are taking place (e.g., by setting $d_{\mathrm{L}} = 0$). We analytically derive approximate node degree distributions.

In Section 6.2 we model the evolution of node indegree and outdegree as a *Degree Markov Chain* (Degree MC). This model is more accurate than the analytical one since it assumes positive loss and makes no assumptions on initialization. We show that when using parameters corresponding to the assumptions in Section 6.1 ($d_{\mathrm{L}} = 0$, constant $d(u) + 2\,d_{\mathrm{in}}(u)$ for all u), the resulting degree distributions are close to the ones obtained analytically.

In Section 6.3 we propose guidelines for selecting protocol parameters s and d_{L}. We show that *S&F* can operate with small views– constant or logarithmic in system size.

Finally, in Section 6.4 we compute the stationary distribution of the Degree MC and show that the protocol preserves M2 - Load Balance.

6.1 Analytically Approximating Degree Distributions without Loss

We start from defining a node *sum degree*:

Definition 6.1 (Sum Degree)**.** *Define $ds(u) = d(u) + 2\,d_{\mathrm{in}}(u)$ to be a* sum degree *of u.*

In this analysis we assume that protocol actions are atomic (no loss), that all views are initialized so that for each u, $ds(u) = d_m$ for some even $d_m \leq s$, and that no edge duplications or deletions are taking place (e.g., by setting $d_{\mathrm{L}} = 0$).

The following proposition shows that sum degrees are preserved by the protocol under the above assumptions.

Lemma 6.2. *If there is no loss, the initial state is chosen so that for some u and some even $d_m \leq s$, $ds(u) = d_m$ and for all v, $ds(v) \leq s$, and $d_{\mathrm{L}} = 0$, then $ds(u) = d_m$ is an invariant.*

Proof. From the initialization, and by the protocol properties, $0 \leq d(v) \leq s$ for each v. Thus, since $d_{\mathrm{L}} = 0$, protocol actions do not perform duplication or deletions. From the protocol, actions that do not involve duplications or deletions do not alter sum degrees. □

Lemma 6.3. *If there is no loss, the initial state is chosen so that for each u, $ds(u) = d_m$ for some even $d_m \leq s$, and $d_{\mathrm{L}} = 0$, the expected node indegree and outdegree is $d_m/3$.*

Proof. By basic graph properties, $\mathbb{E}(d(u)) = \mathbb{E}(d_{\mathrm{in}}(u))$. By initialization and by Lemma 6.2, $\mathbb{E}(d(u)) + 2\,\mathbb{E}(d_{\mathrm{in}}(u)) = ds(u) = d_m$. Clearly, only $\mathbb{E}(d_{\mathrm{in}}(u)) = \mathbb{E}(d(u)) = \frac{d_m}{3}$ satisfies the above equations. □

To analyze node degree distributions under the assumptions of no loss and no duplications or deletions, we start from selecting a node u and d_m nodes $v_1, \ldots, v_{d_m}$. We now decide, for each v_i, whether it become an in-neighbor, out-neighbor, or not-a-neighbor of u, while making sure that $ds(u) = d_m$. For a given even outdegree d^* (and the corresponding indegree of $\frac{d_m - d^*}{2}$), the number of different assignments of $v_1, \ldots, v_{d_m}$ to in-neighbor, out-neighbor, or not-a-neighbor of u that achieve this outdegree is at most:

$$a(d) \triangleq \binom{d_m}{d^*}\binom{d_m - d^*}{\frac{d_m - d^*}{2}}.$$

Given u, $v_1, \ldots, v_{d_m}$, and some assignment Λ, denote the number of different membership graphs containing the assigned subgraph by $b(u, v_1, \ldots, v_{d_m}, \Lambda)$. Although the values of $b(u, v_1, \ldots, v_{d_m}, \Lambda)$ are similar for different choices of $u, v_1, \ldots, v_{d_m}$, and Λ, they is not equal, since different assignments leave slightly different degrees of freedom in the assignments of other nodes. In Section 7.2 (Lemma 7.3) we show that under the assumptions of this section, the protocol is equally likely to reach each membership graph satisfying sum degree invariant ($ds(u) = d_m$ for each u). Thus,

$$\begin{aligned} \Pr(d(u) = d^*) &= \Pr\left(d_{\text{in}}(u) = \frac{d_m - d^*}{2}\right) \\ &\approx \frac{a(d^*)}{\sum_{d'=0,2,4,\ldots,d_m} a(d')}. \end{aligned} \qquad (6.1)$$

The only source of imprecision is the slight variation of the remaining degrees of freedom described above. Figure 3 compares these analytical results with a more precise numerical study (Section 6.2). It shows that that the actual outdegree distribution has similar form and variance. Moreover, it can be seen that the degree distributions of *S&F* have lower variance than the binomial distributions with same expectations.

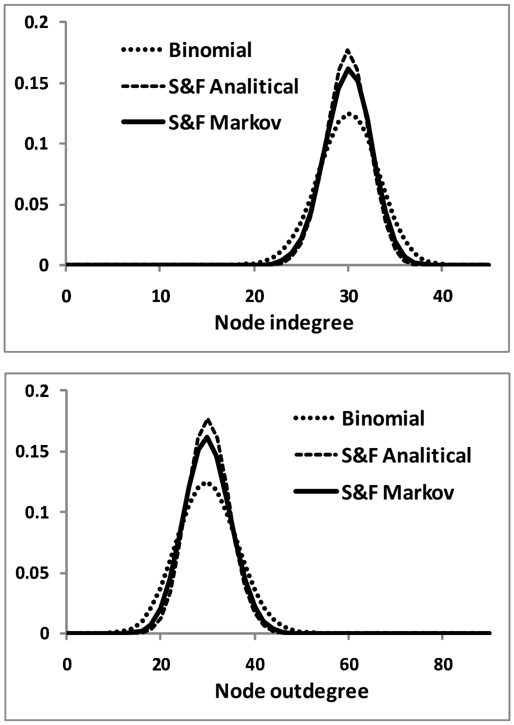


Figure 3: *S&F* node degree distributions (analytical approximation and exact, from Degree MC) and binomial distributions with same expectation. $s = 90$, $d_L = 0$, $\ell = 0$, $ds(u) = 90$ for each u.

6.2 Degree Markov Chain

Allavena [1] analyzed the indegree distribution of a different protocol, with a constant outdegree, assuming no message loss, using a one-dimensional MC. Since in *S&F* both node indegree and outdegree can vary, we construct a two-dimensional *Degree Markov Chain*, where one dimension is indegree and the other is outdegree, reflecting their joint evolution at a single node. We assume that the initial membership graph is weakly connected and that node outdegrees are between d_L and s and are even (*S&F* preserves the latter).

A schematic diagram of the Degree Markov Chain is shown in Figure 4. Note that the state corresponding to an isolated node (zero indegree and outdegree) is disconnected from the rest of the states. In the settings we consider, when the loss is nonzero, $d_L > 0$, so the outdegree cannot decrease to 0. With no loss, we allow $d_L = 0$ but since the initial membership graph is weakly connected, by Lemma 6.2 no node can become isolated.

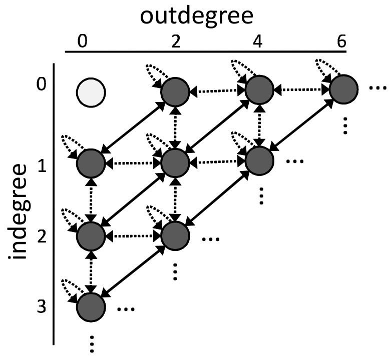


Figure 4: Degree Markov Chain. Dark circles are reachable states and the light circle is an unreachable state. Solid lines correspond to transformations occurring with atomic actions (no loss, duplications, or deletions). Dashed lines correspond to transformations occurring due to loss, duplications, or deletions.

Unfortunately, there is a cycle here: the degree distributions can be learned from the stationary distribution of the MC, but the transition probabilities, in turn, depend on the degree distributions. For example, the probability of a node to receive a message depends on that node's indegree. We therefore search the correct degree distributions iteratively, starting from an arbitrary one, computing the corresponding MC's stationary distribution, and deriving from it the degree distributions, with which we start the next iteration. In each iteration, we compute the MC's stationary distribution numerically, by multiplying the transition matrix by itself until it converges. We stop the computation when the process converges to a MC with matching degree distributions and transition probabilities.

Note that since the sum degree invariant (Lemma 6.2) does not hold with non-atomic actions, sum degrees are not bounded. Considering all possible sum degrees is computationally infeasible. We observed that states with sum degrees close to $3s$ had negligible probabilities under the stationary distribution, so there is not point in computing probabilities for states with higher sum degrees. Therefore, we considered sum degrees to be bounded by $3s$, removing states with higher sum degrees from the MC and replacing edges leading to these states with self-loops.

The resulting degree distributions, for $s = 90$, $d_L = 0$, $\ell = 0$, and $ds(u) = 90$ for each u, shown in Figure 3, have lower variance than that of the binomial distribution. It validates our analysis in Section 6.1, which we use next to set protocol degree thresholds.

6.3 Setting the Thresholds

We first select $\hat{d}$ – the expected outdegree we are interested in without loss. $\hat{d}$ should be chosen based on the application needs (typically $\hat{d} = O(\log n)$ [8]), and, as we see later, on the expected loss rate. Given $\hat{d}$, we now show how to set d_{L} and s so that without loss, the probability of edge duplications and deletions is arbitrarily low, while keeping the expected outdegree close to $\hat{d}$. Suppose we are interested in duplication and deletion probabilities of at most δ, we then look for d_{L} and s satisfying, under no loss, the following conditions: (1) $\mathbb{E}(d(u)) = \hat{d}$, (2) $\Pr(d(u) \leq d_{\mathrm{L}}) < \delta$, and (3) $\Pr(d(u) \geq s) < \delta$. By Lemma 6.3, we set $d_m = 3\hat{d}$. For a given $\delta < 1/2$ we use Equation 6.1 to set

$$d_{\mathrm{L}} = \max_{d'=0,2,4,\ldots,\hat{d} \,:\, \Pr(d(u)\leq d')\leq\delta} d',$$

$$s = \min_{d'=\hat{d},\hat{d}+2,\hat{d}+4,\ldots,d_m \,:\, \Pr(d(u)\geq d')\leq\delta} d'.$$

Since the values of d_{L} and s are discrete, $\Pr(d(u) \leq d_{\mathrm{L}})$ and $\Pr(d(u) \geq s)$ are close but not necessarily equal. Consequently, the resulting expected outdegree may differ from $\hat{d}$ slightly. For example, for $\hat{d} = 30$ and $\delta = 0.01$, d_{L} should be set to 18 and s to 40, resulting in expected outdegree of 30.167. Note that while high δ increases dependencies between nodes' views, setting δ too low decreases the ability of the protocol to fix degree imbalances caused by loss. Typically, $\delta = 0.01$ provides a good balance of keeping low duplication and deletion probabilities with no loss, and fixing degree imbalances under moderate loss.

We conclude that *S&F* satisfies M1 - Small Views property, as even a constant size (in the system size n) views are sufficient for the protocol to function properly.

6.4 Node Degrees with Loss

Figure 5 shows the indegree and the outdegree distributions for several different loss rates and the values $d_{\mathrm{L}} = 18$ and $s = 40$ from the example in Section 6.3.

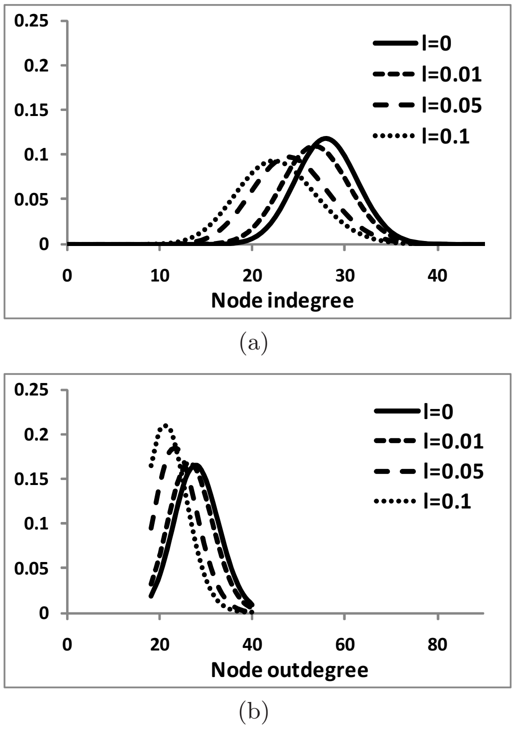
It can be seen that while the average degree decreases with loss, the indegree distribution remains concentrated around the expected degree. Thus, most nodes have similar indegrees and we conclude that the protocol satisfies property M2 - Load Balance.

The next lemma proves what is evident from Figure 5 – that the expected outdegree decreases with increasing loss.

Lemma 6.4. *The expected node outdegree decreases with increasing ℓ.*

PROOF. Assume loss rate ℓ_1 and the corresponding average outdegree d_1 and duplication probability dup_1. Suppose now the loss rate increases to $\ell_2 > \ell_1$. To accommodate higher loss rate, the duplication probability have to increase to $dup_2 > dup_1$, while the deletion probability should not grow. For duplication probability to increase, node outdegrees should reach its lower threshold d_{L} more frequently, and its its upper threshold s at most as frequently as with ℓ_1. This, in turn, implies that expected outdegree decreases. We conclude that in the under loss rate ℓ_2, the expected outdegree $d_2 < d_1$. □

By Lemma 6.4, with increasing loss rate, the expected outdegree approaches its lower bound of d_{L}, the variance of node outdegree decreases (can be observed in Figure 5(b)), and the following observation follows.



(a)

(b)

Figure 5: *S&F* node degree distributions (exact, from Degree MC) for different loss rates $\ell = 0, 0.01, 0.05, 0.1$ ($d_{\mathrm{L}} = 18$, $s = 40$).

Observation 6.5. *The deletion probability decreases with increasing ℓ.*

This is illustrated in Figure 5(b), where the deletion probability is the probability density at the right edge of the curve, as deletions occur only when the outdegree reaches s.

7. UNIFORMITY AND INDEPENDENCE

In this section we analyze the remaining protocol properties of uniformity and independence (M3 – M5). We assume initial graph is weakly connected and the node outdegrees are between d_{L} and s and are even. In Section 7.1 we define a global Markov Chain graph that we use to model protocol actions. In Section 7.2 we prove that with no loss and no duplications or deletions, all membership graphs reachable from the initial graph are equally likely to be reached by the protocol. In Section 7.3 we show that eventually each node id is equally likely to appear in each other node's view. In Section 7.4 we show that the expected fraction of independent entries in views is at least $1 - 2(\ell + \delta)$. Finally, in Section 7.5 we show that the number of actions each node needs to initiate in order to reach a state that is independent of the initial state is bounded by $O(\log n)$ for constant size views and by $O(\log^2 n)$ for logarithmic views.

7.1 The Global Markov Chain Graph

We define $\mathcal{G}(s, d_{\mathrm{L}}, \ell)$ to be the *Global Markov Chain Graph* induced by *S&F* with given s, d_{L}, and ℓ. For simplicity, we omit the parameters and refer to this graph as $\mathcal{G}$. The set of vertices of $\mathcal{G}$, $\mathcal{V}$, contains all the membership graphs that can be reached by *S&F* from any initial weakly-connected membership graph $G(0)$, where all initial node outdegrees are between d_{L} and s and are even (*S&F* preserves the latter). We call vertices in $\mathcal{G}$ *states*, as each vertex represents a global state of the views of all nodes. States G_1 and G_2 are connected by a directed edge (G_1, G_2) if there exists at least one transformation from G_1 to G_2. The weight of the edge, $p(G_1, G_2)$ is the sum of probabilities of all transformations from G_1 to G_2.

Note that some states in $\mathcal{G}$ might be partitioned membership graphs, e.g., when some node has no incoming edges and all its outgoing edges are self-edges. We remove partitioned states from $\mathcal{G}$ and replace the edges leading to them by self-loops. In Section 7.4 we show sufficient conditions for making the probability of reaching such partitioned membership graphs arbitrarily small. When these conditions do not hold, e.g., when the loss rate is 100%, the analysis in this section is not applicable. There are also states that are unreachable from other states: the states where all views are full ($d(u) = s$ for each u). We assume that the initial state is not among these states and thus remove them from $\mathcal{G}$.

After removing partitioned and unreachable states, we get that the vertices of G are exactly the weakly-connected membership graphs where node outdegrees are between d_{L} and s (but not all equal s) and are even. Note that each state in $\mathcal{G}$ has a self-loop edge corresponding to *self-loop transformations*, that occur as a result of actions where one of the selected view entries is empty so the action has no effect on the views.

The proof of the following lemma appears in [15].

Lemma 7.1. *When $0 < \ell < 1$, $\mathcal{G}$ is strongly connected.*

Lemma 7.1 implies that from any initial state, any state in $\mathcal{G}$ can be reached by a sequence of *S&F* transformations.

Lemma 7.2. *The Markov Chain on $\mathcal{G}$ has a unique stationary distribution π.*

PROOF. Clearly, $\mathcal{G}$ is finite. By Lemma 7.1 it is irreducible. It is aperiodic (meaning that the greatest common denominator of the lengths of directed paths connecting any two nodes in $\mathcal{G}$ is 1) since each state in $\mathcal{G}$ has a self-loop edge. From the above, the Markov Chain is ergodic, and, by the fundamental theorem of the theory of Markov Chains, has a unique stationary distribution. □

Definitions.

Steady state is a random state distributed according to π.

Expected outdegree d_{E} is the expected node outdegree in the steady state. It is immediate that $d_{\mathrm{E}} \geq d_{\mathrm{L}}$.

Expected independence α is the expected fraction of independent entries in views in the steady state.

7.2 Stationary Distribution with No Loss

We now complete the analysis of Section 6.1, by proving that with no loss and when for each u, $ds(u) \leq s$ and is even, the stationary distribution over all reachable states in $\mathcal{G}$ is uniform. As we assume no loss, there is no need to compensate for it using duplications, so we set $d_{\mathrm{L}} = 0$. It is easy to see that in the above setting, no duplications or deletions take place. Observe that by Lemma 6.2, *S&F* preserves the sum degree of each node. Let $\bar{\mathbf{ds}} = (ds(u), ds(v), \ldots)$ be the vector of initial node sum degrees. For the sake of the analyses in this section, we define $\mathcal{G}_{\bar{\mathbf{ds}}}$ to be the subgraph of $\mathcal{G}$ where all states satisfy a given degree sum vector $\bar{\mathbf{ds}}$. Then, $\mathcal{G}_{\bar{\mathbf{ds}}}$ is the MC graph induced by *S&F* under the above assumptions, where nodes have sum degrees according to $\bar{\mathbf{ds}}$.

In [15], we prove the following lemma, which asserts that the stationary distribution of the MC on $\mathcal{G}_{\bar{\mathbf{ds}}}$ is uniform. The proof is basically an adaptation of the proof in [19] to *S&F*.

Lemma 7.3. *The stationary distribution of the MC on $\mathcal{G}_{\bar{\mathbf{ds}}}$ is a uniform distribution over all states in $\mathcal{G}_{\bar{\mathbf{ds}}}$.*

7.3 Proving Uniformity (M3)

We now return to the general case, where loss may occur. We show that property M3 - Uniform Sample holds, with the exception that the probability that u's view contains its own id may be different (higher) than the uniform probability to contain any other id $v \neq u$.

Lemma 7.4. *In the steady state, for each u, u's view contains each $v \neq u$ with equal probability.*

PROOF. Consider two arbitrary nodes u and v. Denote by $\mathcal{G}_{(u,v)}$ the set of states in $\mathcal{G}$ that contain edge (u, v). As $\mathcal{G}$ includes all weakly-connected membership graphs where $d_{\mathrm{L}} \leq d(u') \leq s$ for each u', and since all nodes behave exactly the same way, by symmetry, for all u, v, w, z, such that $u \neq v$ and $w \neq z$, the subgraph spanned by $\mathcal{G}_{(u,v)}$ is isomorphic to the subgraph spanned by $\mathcal{G}_{(w,z)}$. Thus, in $\mathcal{G}$'s stationary distribution π, the probability of being in one of the states in $\mathcal{G}_{(u,v)}$ equals the probability of being in one of the states in $\mathcal{G}_{(w,z)}$. From here, every node $v \neq u$ has the same positive probability to appear in u's view. □

7.4 Proving Spatial Independence (M4)

We next analyze property M4 - Spatial Independence and show that in the steady state, the expected fraction of independent entries in all views, α, can be bounded from below by some positive constant.

In this section, we restrict the initial state, and assume that initially, the fraction of independent entries in views is at least 2/3. We show that under moderate loss, this fraction converges to a much higher value. Thus, α remains greater than 2/3.

Assumption 7.5. $\alpha \geq 2/3$.

Note that due to Assumption 7.5 our analysis is not applicable for high loss rates, where α might become too low. Nevertheless, since our analysis is not tight, we speculate that the protocol may work well also with α below 2/3. The exact dependence of α on the loss rate will become evident in the analysis below.

Observe that spatial independence decreases only when the protocol performs duplication, creating dependent entries in views of immediate neighbors. Recall that δ is the duplication probability of the protocol with no loss. We get the following bound on duplications:

Lemma 7.6. *The duplication probability during non-self-loop transformations is at most $\ell + \delta$.*

PROOF. In the steady state, the probability of duplication equals ℓ plus the probability of deletion. By Observation 6.5, for $\ell > 0$, the probability of deletion decreases below δ. The lemma follows. □

The following analysis shows that the expected fraction of independent entries in views is bounded from below by $1-2(\ell+\delta)$. Note that typically, both ℓ (see [22, 4]) and δ (see Section 6) are in the order of 1%, hence the vast majority of view entries are expected to be independent.

The following lemma is proven in [15]. It coarsely bounds the probability for a dependent view entry that u sends to return to u in the future. By slight abuse of terminology, we use the term *dependent entry* to refer to a particular instance of an id that was created by duplication. The dependent entry is created in some view entry of u, and later may be sent to other nodes and reside in their views. In this lemma we ignore the possibility that a dependent entry is duplicated again, and account for this in a later lemma.

Lemma 7.7. *Suppose u sends a dependent entry to one of its neighbors. In the steady state, the probability for this entry to be sent back to u in the future is at most $1/2$.*

Intuitively, the lemma follows from the fact that u's neighbors have many additional neighbors, and thus the id is more likely to travel away from u than to return.

Lemma 7.8. *In the steady state, the expected fraction of independent entries in views is bounded from below: $\alpha \geq 1-2(\ell+\delta)$.*

Proof. We analyze the expected time a nonempty entry in a view is independent. Since the protocol is memoryless, we use a simple *Dependence Markov Chain* to model the state of the entry, which can be either "dependent" or "independent".

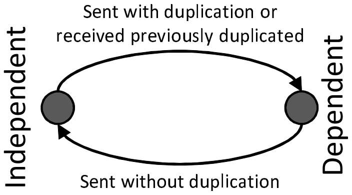


Figure 6: Dependence Markov Chain.

We consider non-self-loop transformations corresponding to actions initiated by a random node u and bound the transition probabilities between these states. We then compute the stationary distribution of the Dependence MC, shown in Figure 6, and derive from it the bound on the expected time a nonempty entry in a view is independent. We ignore self-loop transformations since they do not cause any change in views and thus do not alter the dependence state of any entry.

We start with computing probability of going from the independent to the dependent state. By Proposition 5.1 each entry has the same probability to be involved in a transformation. Thus, by Lemma 7.6, the probability of a entry to become dependent during a non-self-loop transformation is at most $\ell+\delta$. By Lemma 7.7, the probability of receiving previously duplicated entry in the future is at most $1/2$. Thus, in the steady state, the arrival rate of the returning dependent entries is at most half of the rate of creation of the new dependent entries. Summing up, the probability of going from the independent to the dependent state is at most $(1+\frac{1}{2})(\ell+\delta) = \frac{3}{2}(\ell+\delta)$.

We now bound the probability of going from the dependent to the independent state. An action removes a dependent entry from a view if (1) the target node is different from the action initiator, and (2) the entry is not duplicated again. By Lemma 7.6, the probability of (2) is bounded by $1-(\ell+\delta)$. We next bound the probability of (1).

Let β be the probability of an entry to be a *self-edge*, i.e., $u.lv[i] = u$. The most likely scenario for creating a self-edge in u's view is: (1) u creates two parallel edges (v,u) by initiating two actions involving one of its out-neighbor v (in both u sends a message to v which is not lost or deleted), where the first action performs duplication so that v's id remains in u's view; then, (2) v initiates an action involving both of these parallel edges (v,u), send message $[v,u]$ to u and the message is not lost or deleted. Since the probability of (2) is at most $1/2$ by Lemma 7.7, we conclude that at most half of the dependent entries are self-edges. Since we assumed $\alpha \geq 2/3$ (Assumption 7.5), the probability β of a random view entry to be a self-edge is at most $\frac{1}{3}\cdot\frac{1}{2}=\frac{1}{6}$. Summing up, the probability of going from the dependent to the independent state is at least $(1-\beta)(1-(\ell+\delta)) = \frac{5}{6}(1-(\ell+\delta))$.

Thus, an entry is expected to spend at most $\frac{1}{\frac{5}{6}(1-(\ell+\delta))}$ out of $\frac{1}{\frac{3}{2}(\ell+\delta)}+\frac{1}{\frac{5}{6}(1-(\ell+\delta))}$ transformations in the dependent state.

$$\frac{\frac{1}{\frac{5}{6}(1-(\ell+\delta))}}{\frac{1}{\frac{3}{2}(\ell+\delta)}+\frac{1}{\frac{5}{6}(1-(\ell+\delta))}} = \frac{\frac{\frac{6}{5}}{(1-(\ell+\delta))}}{\frac{\frac{2}{3}(1-(\ell+\delta))+\frac{6}{5}(\ell+\delta)}{(\ell+\delta)(1-(\ell+\delta))}}$$
$$= \frac{\frac{6}{5}(\ell+\delta)}{\frac{2}{3}+\frac{8}{15}(\ell+\delta)} = \frac{\ell+\delta}{\frac{5}{9}+\frac{4}{9}(\ell+\delta)} \leq 2(\ell+\delta).$$

The lemma follows. □

Connectivity conditions. A sufficient condition for a membership graph to be weakly connected is that each node has at least three independent out-neighbors [9]. Although we do not know the exact distribution of the number of independent ids in views, since the loss (and hence the duplications) are uniform and independent, we speculate that the number of independent ids in node views is distributed similarly to node outdegree but with lower expectation ($\alpha\, d_{\mathrm{E}}$ instead of d_{E}). That is, the number of independent ids in a view is distributed close to a binomial distribution with expectation of at least $\alpha\, d_{\mathrm{L}}$. Thus, for any given probability ϵ and loss rate ℓ, we can find the minimal d_{L} guaranteeing that the probability of a node to have less than 3 independent neighbors is at most ϵ. E.g, for $\ell=\delta=1\%$, and $\epsilon=10^{-30}$, d_{L} should be set to at least 26.

7.5 Proving Temporal Independence (M5)

We next analyze M5 - Temporal Independence. Consider a random initial state $G(0)=\tilde{G}$ chosen from π. Clearly, the state $\tilde{G}(1)$ after one transformation is highly dependent on $G(0)$. However, as more transformations are performed, the dependence between $\tilde{G}(i)$ and $G(0)$ decreases. For a given ϵ, we would like to find the minimum time $\tau_\epsilon(\mathcal{G})$ such that for all subsets of states S,

$$|\Pr[\tilde{G}(\tau_\epsilon(\mathcal{G})) \in S|\, G(0)=\tilde{G}] - \pi(S)| < \epsilon.$$

That is, after $\tau_\epsilon(\mathcal{G})$ transformations, the membership graph is ϵ-independent of the initial graph. Note that this does not bound the MC's mixing time, since we start from a random $\tilde{G}$, distributed according to π. We do this in order to avoid starting from rare pathological states where view entries are much more dependent than expected. Fortunately, as we showed in Section 7.4, in an expected state the fraction of dependent entries is bounded by a small constant. Thus, the total weight of such pathological states under π is negligible.

For the sake of this analysis, we assume that there are exactly n nodes in all states in $\mathcal{G}$ and that $s \ll \sqrt{n}$. We derive (in [15]) the expected conductance – a generalization of graph expansion around the expected state – of $\mathcal{G}$ from three properties: (1) each transition from each state is induced by two entries selected uniformly at random in a view of a random node; (2) both of these transitions are not self-loops (due to empty view entries) with probability $\frac{d_{\mathrm{E}}(d_{\mathrm{E}}-1)}{s(s-1)}$; and (3) the expected fraction of independent entries in views is bounded from below by α, hence different transitions involving independent view entries lead to different states, independently of other transitions, with probability of at least α. We then use standard techniques typically used to deduce the mixing time from conductance to show (also in [15]):

Lemma 7.9. *Assuming $s \ll \sqrt{n}$,*

$$\tau_\epsilon(\mathcal{G}) \;\leq\; \frac{16\,s^2(s-1)^2}{{d_{\mathrm{E}}}^2(d_{\mathrm{E}}-1)^2\,\alpha^2}\left(n\,s\cdot\log(n)+\log\frac{4}{\epsilon}\right).$$

Note that for zero loss, $\alpha = 1$, and temporal independence is achieved in $O(n\,s\log n)$ transformations. That is, after each node initiates $O(s\log n)$ actions in expectation, the views of all nodes are independent of the initial state. For logarithmic view sizes this translates to $O(\log^2 n)$ time until the dependence on the initial state becomes arbitrarily low. For a positive but moderate loss, α remains a constant bounded away from 0, and the time it takes to achieve temporal independence increases by a constant factor.

8. CONCLUSIONS

We formalized the desired properties of distributed membership service: small local views, bounded number of node neighbors, uniformity of views, and their low correlation with past and neighbors' views. We proposed a formal model for studying membership graph evolutions with non-atomic protocol actions. We presented a simple and practical membership protocol, *S&F* and showed that it provides all the desired properties of a membership service. This is the first analysis of a membership protocol in the presence of message loss that we are aware of. It might be interesting to apply our methodology in order to analyze additional gossip-based protocols under message loss.

Acknowledgments

We are grateful to Fabian Kuhn for stimulating discussions on the expansion of random graphs.

9. REFERENCES

[1] A. Allavena. *On the correctness of gossip-based membership protocols*. PhD thesis, Cornell University, 2006.

[2] A. Allavena, A. Demers, and J. E. Hopcroft. Correctness of a gossip based membership protocol. In *PODC*, pages 292–301, 2005.

[3] C. Avin, M. Koucký, and Z. Lotker. How to explore a fast-changing world (cover time of a simple random walk on evolving graphs). In *ICALP*, pages 121–132, 2008.

[4] O. Bakr and I. Keidar. Evaluating the running time of a communication round over the internet. In *PODC*, pages 243–252, 2002.

[5] Z. Bar-Yossef, R. Friedman, and G. Kliot. RaWMS - Random Walk based Lightweight Membership Service for Wireless Ad Hoc Networks. In *ACM MobiHoc*, pages 238–249, 2006.

[6] E. Bortnikov, M. Gurevich, I. Keidar, G. Kliot, and A. Shraer. Brahms: byzantine resilient random membership sampling. In *PODC*, pages 145–154, New York, NY, USA, 2008. ACM.

[7] Y. Busnel, M. Bertier, and A.-M. Kermarrec. Bridging the Gap between Population and Gossip-based Protocols. Research Report RR-6720, INRIA, 2008.

[8] P. T. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, and A.-M. Kermarrec. Lightweight probabilistic broadcast. *ACM TOCS*, 21(4):341–374, 2003.

[9] T. I. Fenner and A. M. Frieze. On the connectivity of random m-orientable graphs and digraphs. *Combinatorica*, 2(4):347–359, 1982.

[10] M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382, Apr. 1985.

[11] A. J. Ganesh, A.-M. Kermarrec, and L. Massoulie. SCAMP: Peer-to-Peer Lightweight Membership Service for Large-Scale Group Communication. In *Networked Group Communication*, pages 44–55, 2001.

[12] D. Gavidia, S. Voulgaris, and M. van Steen. Epidemic-style monitoring in large-scale sensor networks. Technical Report IR-CS-012, Vrije Universiteit, Netherlands, March 2005.

[13] C. Gkantsidis, M. Mihail, and A. Saberi. Random walks in peer-to-peer networks. In *IEEE INFOCOM*, 2004.

[14] J. Gray. Notes on data base operating systems. In *Advanced Course: Operating Systems*, pages 393–481, 1978.

[15] M. Gurevich and I. Keidar. Correctness of gossip-based membership under message loss. Technical Report CCIT Report #732, Department of Electrical Engineering, Technion, 2009.

[16] M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, and M. van Steen. Gossip-based peer sampling. *ACM Trans. Comput. Syst.*, 25(3):8, 2007.

[17] C. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in unstructured peer-to-peer networks. In *ICS*, pages 84–95, 2002.

[18] P. Mahlmann and C. Schindelhauer. Peer-to-peer networks based on random transformations of connected regular undirected graphs. In *SPAA*, pages 155–164, 2005.

[19] P. Mahlmann and C. Schindelhauer. Distributed random digraph transformations for peer-to-peer networks. In *SPAA*, pages 308–317, New York, NY, USA, 2006. ACM.

[20] L. Massoulie, E. L. Merrer, A.-M. Kermarrec, and A. J. Ganesh. Peer Counting and Sampling in Overlay Networks: Random Walk Methods. In *PODC*, pages 123–132, 2006.

[21] R. Melamed and I. Keidar. Araneola: A scalable reliable multicast system for dynamic environments. *J. of Parallel and Distributed Computing*, 68(12):1539 – 1560, 2008.

[22] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson. The end-to-end effects of internet path selection. *SIGCOMM Comput. Commun. Rev.*, 29(4):289–299, 1999.

[23] S. Voulgaris, D. Gavidia, and M. van Steen. CYCLON: Inexpensive Membership Management for Unstructured P2P Overlays. *J. of Network and Systems Management*, 13(2):197–217, July 2005.