SIAM J. COMPUT.
Vol. 39, No. 8, pp. 3830–3859

CORRECTNESS OF GOSSIP-BASED MEMBERSHIP UNDER MESSAGE LOSS*

MAXIM GUREVICH† AND IDIT KEIDAR‡

Abstract. Due to their simplicity and effectiveness, gossip-based membership protocols have become the method of choice for maintaining partial membership in large peer-to-peer 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 delayless 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.

Key words. peer-to-peer, membership, gossip, random sampling

AMS subject classifications. 68M14, 68M15, 68W15, 68W40

DOI. 10.1137/090769752

1. Introduction.

Large-scale dynamic systems are now being deployed in many places, including peer-to-peer networks over the Internet, in data centers, and in 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 composed of unreliable components, where nodes can fail 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 [13, 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 properties 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"

*Received by the editors September 2, 2009; accepted for publication (in revised form) September 10, 2010; published electronically December 14, 2010. A preliminary version of this paper appeared in the proceedings of the Twenty-Eighth Annual ACM Symposium on the Principles of Distributed Computing (PODC) [21]. This work was partially supported by the Technion Security Research Fund and by the Israeli Ministry of Industry, Trade, and Labor.

http://www.siam.org/journals/sicomp/39-8/76975.html

†Department of Electrical Engineering, Technion, Haifa 32000, Israel. Current address: Yahoo! Research, 4401 Great America Parkway, Santa Clara, CA 95054 (maximg@yahoo-inc.com). This author's work was partially supported by the Eshkol Fellowship of the Israeli Ministry of Science.

‡Department of Electrical Engineering, Technion, Haifa 32000, Israel (idish@ee.technion.ac.il).

for a membership service is to choose view entries independently of each other (we call this *spatial independence*) and uniformly at random [13, 29, 9]. Indeed, such choices result in an expander graph, with good connectivity, robustness, and low diameter [15], 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 sufficiently connected initial topology resulting from joins, leaves, and failures.

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 [25, 18, 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 [17, 13, 2, 34, 23]. In such protocols, nodes exchange ("gossip about") ids from their views with their neighbors and use this information to update their views (see section 3.1). 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 [13, 23], and certain variants of gossip were proved 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, 26, 27, 11]. 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 concurrently 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 [20] or node failures.

Our main goal in this paper is to bridge the gap between protocols that work well in practice but are not amenable to formal analysis to others that admit analysis but make overly conservative assumptions that limit their practical applicability. We propose a methodology for designing and analyzing protocols with nonatomic actions, and apply it to design the first protocol that at the same time (1) is practical, in that it can be implemented in fault-prone networks without any bookkeeping, (2) is amenable to formal analysis, and (3) does not induce spatial dependencies.

In section 4, we present a model for studying gossip-based membership without atomicity assumptions. We follow [26, 27] 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 few 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 views that are logarithmic in system size. It further shows that the distribution of node degrees is very well balanced—close to the binomial distribution. We then analyze degree evolution of joining and leaving nodes and the time it takes to integrate new nodes and to remove ids of left/failed ones from views of other nodes.

In section 7 we study the distribution of membership graphs to which the protocol evolves (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 about twice as fast as the loss rate. As the loss is typically in the order of 1% [32, 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, the system evolves to a state whose dependence on G can be made arbitrarily small.

In summary, our key contribution is in formally analyzing a protocol that can work in the real world; this includes the following:

(i) We spell out the desired properties of membership protocols that maintain small views.

(ii) We provide a model for studying membership graph evolution with nonatomic protocol actions.

(iii) We present a practical membership protocol, S&F, which is amenable to formal analysis.

(iv) In the absence of message loss, S&F provides all the desired properties of a membership service.

(v) 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, $u.lv$, of s node ids, also denoted $u.lv[1 \dots s]$. We say that u is an *in-neighbor* of v, and that v is an *out-neighbor* of u, if $v \in u.lv$. We denote such a view entry by (u, v). For simplicity, we allow a view to contain duplicate ids and account for them later as dependencies. 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 that 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 that u has.

We now formalize the desirable properties of a distributed membership service. Later, in section 4, we define a set of "building blocks" for distributed protocols that implement such a 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 the following property.

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 [13]. Other applications work with constant-size views [29]. 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 [7], 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 sufficiently connected 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 (i.i.d.) uniform sample of the nodes. The next two properties stipulate that views should converge to i.i.d. uniform ones, from any state.

PROPERTY M3 (uniform sample). *Starting from any initial state, eventually, for each* u, v, w,

$$\Pr(v \in u.lv) = \Pr(w \in u.lv).$$

Uniformity, by itself, does not imply independence among view entries of the same node or of different nodes at the same time. Therefore Property M3 does not subsume Property M2: Property M3 means that every id eventually has the same likelihood of appearing in any given view entry. However, Property M3 does not preclude dependencies among distinct entries (e.g., duplicate ids in a view) at a given 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 $u.lv[i]$ and $v.lv[j]$ are *independent* of each other if

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

By slight abuse of terminology, we simply label edges in a membership graph as dependent without specifying what edges they depend on. We label edges as follows: (1) All self-edges ($u.lv[i] = u$) are *dependent*.[1] (2) For $v = u$ or $v \in u.lv$, if $u.lv[i]$ is not independent of $v.lv[j]$ for some j, then we say that one of $u.lv[i]$ or $v.lv[j]$ is

[1]Even perfect i.i.d. sampling can produce self-edges. However, since this happens extremely infrequently, we conservatively consider all self-edges to be dependent.

dependent. In case of dependencies among several edges, all but one of these edges are considered dependent. Intuitively, these edges all convey similar information, so we can choose one of them as representative and discount the others. 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* $u.lv[i]$ *is nonempty, the probability that* $u.lv[i]$ *is independent is bounded from below by a constant independent of* n.

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*), the number of actions the protocol needs to take in order to reach a state that is independent of the initial state is bounded from above.*

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.

3.1. 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, 26, 27].

The ids sent. Allavena, Demers, and Hopcroft [2] identified two crucial components for a good membership protocol: In a *reinforcement* component, a node adds its own id to another node's view. Reinforcement leads to a uniform representation of nodes in other nodes' views and fixes any nonuniformity that might have been caused by bad initial views or churn. In a *mixing* component, a node adds to its view an id from another 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, 27], Lpbcast [13] uses push for both. We do the same in this paper. Occasionally, due to the common reinforcement-by-push and mixing-by-pull association, push-only and pull-only protocols are deemed impractical [23]; however, these are actually reinforcement-only and mixing-only protocols that are impractical. A practical optimization, made in many protocols, e.g., [13, 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, 26, 27], 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., [13, 2], keep the sent ids, thus inducing dependence between neighbor views. Those that delete the sent ids, e.g., shuffle [1, 27] and flipper [26], are unable to withstand message loss or node failures since the system gradually loses more and more ids. In fact, by design, these protocols work only with a static membership and provide no means for joining or leaving the system. Jelasity et al. [23] 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 ids that are actually lost 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 (nonlocal) approach is to use random walks (RWs) (on the membership graph) to obtain new ids for local views [19, 5, 28]. However, RWs are disadvantageous in our setting. First, since a single RW involves multiple id exchange steps, the probability of a successful RW under message loss degrades exponentially with the length of the random walk. Second, an RW's correctness depends on the graph topology. Unlike gossip, where views are updated after every step (regardless of the graph topology), an RW explicitly stops at some point and then takes a sample. If the actual topology is different from the assumed one, then that sample may be far from uniform [19]. Third, 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 [7] uses fast evolving local views, which might be nonuniform, 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.

3.2. Markov chains. Here we provide a brief introduction into the theory of MCs. For more details please refer to any standard textbook on the subject, e.g., [31, 8].

A *Markov chain* on a finite state space $\mathcal{U}$ is a stochastic process in which states of $\mathcal{U}$ are visited successively. The MC is specified by a $|\mathcal{U}| \times |\mathcal{U}|$ *probability transition matrix* P. P is a *stochastic matrix*, meaning that every row x of P specifies a probability distribution P_x on $\mathcal{U}$. P induces a directed graph G_P on $\mathcal{U}$ with nonnegative edge weights. There is an edge $x \to y$ in the graph if $P(x, y) > 0$ and the corresponding weight is $P(x, y)$.

The MC is called *ergodic* if it satisfies two conditions: (1) it is *irreducible*, meaning that the graph G_P is strongly connected, and (2) it is *aperiodic*, meaning that the greatest common denominator of the lengths of directed paths connecting any two nodes in G_P is 1.

Each step t of the MC induces a probability distribution p_t on the state space $\mathcal{U}$. The initial distribution is p_0. Successive distributions are given by the recursive formula: $p_t = p_{t-1}P$. Therefore, $p_t = p_0 P^t$. A fundamental theorem of the theory of MCs (a.k.a. ergodic theorem) states that if an MC is ergodic, then *regardless* of the initial distribution p_0, the sequence of distributions $p_0, p_1, p_2, \ldots$ is guaranteed to converge to the unique *stationary distribution* π such that $\pi P = \pi$. That is, $||p_t - \pi|| \stackrel{t\to\infty}{\longrightarrow} 0$.

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 $v \in u.lv$, with the multiplicity equal to the multiplicity of v in $u.lv$. 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 an MC $\tilde{G}(0), \tilde{G}(1), \ldots$, where $\tilde{G}(i)$ represents the distribution of the membership graphs after the ith 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. Assuming that the initial membership graph is weakly connected, 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 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 ids 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, 26, 6, 27, 11], assumed atomic actions, with no overlap in time. However, guaranteeing atomicity of multistep 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 [20, 16], even if the nodes themselves are synchronous (Theorem I0 in [12]).

We allow communication to be asynchronous but assume that the nodes are loosely synchronized among themselves, so that they may all independently invoke actions at a similar rate.

Modeling loss with nonatomic actions. Due to message loss, with some probability a sent message is not delivered to its destination. We assume that this probability is unknown to the protocol and that the sender cannot detect that the message it sent was lost, so it cannot retransmit the message. This means that in a multistep action, each step is executed with probability ≤ 1, given that the previous step was executed (except for the first step, which is executed with probability 1).

In this paper we restrict our analysis to uniform loss. We assume a message is lost with probability ℓ, identical for all messages, and independent of other messages.

While nonuniform loss occurs in practice [33], it is more difficult to model and analyze. Thus, similarly to other works dealing with protocol analysis under message loss (e.g., [22, 24]), we resort to the uniform i.i.d. loss model.

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 nonoverlapping. The protocol running at each node is shown in Figure 5.1 (u.a.r. stands for uniformly at random). Each node u maintains a view $u.lv$—an array of size s, where $s \geq 6$ is even.[2] In order to overcome loss (nonatomic actions), the protocol is parametrized by a threshold $0 \leq d_{\mathrm{L}} \leq s-6$ that sets a lower bound on node outdegree. The gap between d_{L} and s makes the outdegree flexible enough for the protocol to be effective.

1: **function** S&F-InitiateAction$_u$()
2: select $1 \leq i \neq j \leq s$ u.a.r.
3: $v \leftarrow u.lv[i]$
4: $w \leftarrow u.lv[j]$
5: **if** $v \neq \perp$ AND $w \neq \perp$ **then**
6: send $[u, w]$ to v
7: **if** $d(u) > d_{\mathrm{L}}$ **then**
8: $u.lv[i] \leftarrow \perp$
9: $u.lv[j] \leftarrow \perp$

1: **function** S&F-Receive$_u(v_1, v_2)$
2: **if** $d(u) < s$ **then**
3: select i u.a.r. so that $u.lv[i] = \perp$
4: select j u.a.r. so that $u.lv[j] = \perp$
5: $u.lv[i] \leftarrow v_1$
6: $u.lv[j] \leftarrow v_2$

FIG. 5.1. *The S&F protocol at node u.*

A joining node has to know at least d_{L} ids of live nodes before engaging in the protocol. A node can obtain these ids by copying another node's view, or, in case of reconnection, by probing previously seen ids. We conservatively require the minimal degree of d_{L} to guarantee weak connectivity of the membership graph with high probability. Nodes that wish to leave the system do not need to take any explicit actions; they simply stop participating in the protocol.

A protocol step at node u works as follows: The node selects two different entries i and j in its view uniformly at random. If either of them is empty, nothing happens and the views of all the nodes remain unchanged. If both $v = u.lv[i]$ and $w = u.lv[j]$ are nonempty, then u performs the following steps: (1) it sends to v a message including its own id and w, and (2) it clears both entries i and j in its view, unless $d(u) \leq d_{\mathrm{L}}$, 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*. Figures 5.2(a) and 5.2(b) show 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 time). Figure 5.2(c) shows the effect of duplication at the sender, and Figure 5.2(d) illustrates message loss or deletion at the receiver. The id of a node that fails/leaves remains in some views of live nodes for some time, but then disappears from all views during the normal course of the protocol, as every message sent to this node causes its id to be deleted from the sender's view (except if duplicated).

[2]We use $s \geq 6$ for proving reachability of every membership graph from every other graph in Lemma A.3. However, this may not be a necessary condition for reachability.

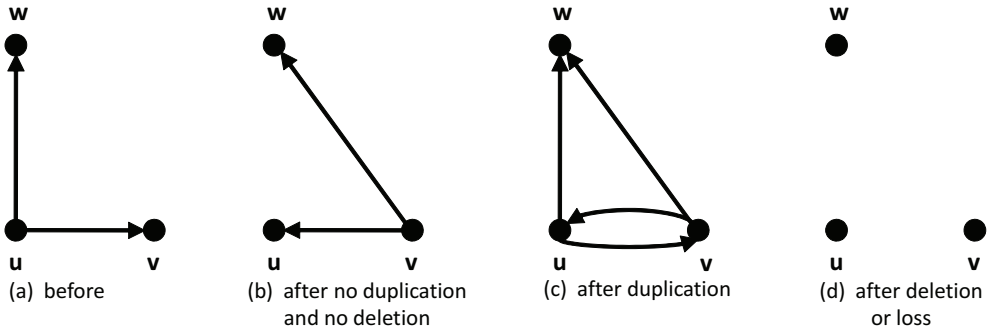


FIG. 5.2. *Possible outcomes of a transformation of S&F, initiated by u sending message* $[u, w]$ *to v, and v performing the receive step.* (a) *Before the transformation. Possible states after the transformation, where* (b) $d(u) > d_{\mathrm{L}}, d(v) < s$, *message delivered;* (c) $d(u) = d_{\mathrm{L}}, d(v) < s$, *message delivered;* (d) $d(u) > d_{\mathrm{L}}$, *and* $d(v) = s$ *or message lost.*

It is easy to see that the protocol satisfies the following invariant.

Observation 5.1. Every node's outdegree is at any time between d_{L} and s and is even.

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 replication since it increases dependencies among ids in the same 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 at the same frequency at all nodes. The next proposition follows immediately.

PROPOSITION 5.2. *The probability for every node u and every two entries in u's view to be chosen in an action is the same.*

Optimizations. One could modify the S&F protocol to make it more efficient by incorporating some lessons from the substantial existing experience with practical membership protocols. Examples include the following: (1) Instead of removing sent ids from the view, the protocol could only mark them for deletion and could then use undeletion instead of duplication; (2) instead of discarding received ids when the view is full, the protocol could replace some existing view entries with new ids; (3) more than two ids could be sent in a message. However, since such optimizations would make the protocol harder to analyze, we opted to avoid them and leave optimizations to future work.

6. Node degree analysis and setting degree thresholds. In this section we show that S&F satisfies Properties M1 (small views) and M2 (load balance) defined in section 2. We assume that $n \gg s$. In the examples in this section, view sizes are up to 100, and n is assumed to be in the order of thousands or more. As long as n is sufficiently large, for fixed s and d_{L}, our results are independent of n.

In this section we analyze the in- and outdegree distributions of a single node in the *steady state*. Steady state is the expected membership graph to which the protocol converges after sufficiently many actions. We formally define, show existence, and analyze the properties of the steady state in section 7. Since we analyze the steady state, we assume the churn ceases for the period we analyze.

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

In section 6.2 we remove the additional assumptions and 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 weaker 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 those 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.

In section 6.4 we compute the stationary distribution of the degree MC and show that the protocol preserves Property M2 (load balance).

Finally, in section 6.5 we analyze the time it takes to integrate a joining node into the system and to remove ids of a left/failed node from views.

6.1. Analytically approximating degree distributions without loss. We start by 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. By Observation 5.1, $0 \leq d(v) \leq s$ 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* $d_{\mathrm{L}} = 0$, *then the average node indegree and outdegree are* $d_m/3$.

Proof. We define the average of function f over the set of nodes as follows: $\mathrm{avg}(f(u)) = \frac{1}{n}\sum_u f(u)$. Since total indegrees and outdegrees are both equal to the number of edges, $\mathrm{avg}(d(u)) = \mathrm{avg}(d_{\mathrm{in}}(u))$. By Observation 5.1 and Lemma 6.2,

$\text{avg}(d(u)) + 2\,\text{avg}(d_{\text{in}}(u)) = ds(u) = d_m$. Clearly, only $\text{avg}(d_{\text{in}}(u)) = \text{avg}(d(u)) = \frac{d_m}{3}$ satisfies the above equations. □

We now analyze node degree distributions of a single node under the assumptions of no loss and no duplications or deletions. Suppose that we want to select neighbors for each node so that the sum degree of each node is d_m. We start with all views being empty and select an arbitrary node u and d_m arbitrary nodes $v_1, \ldots, v_{d_m}$ to be potential neighbors of u. We now decide, for each v_i, whether it becomes an in-neighbor, out-neighbor, or not-a-neighbor of u, while making sure that $ds(u) = d_m$. For a given even outdegree $d^* \in [0, d_m]$ (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 $b(u, v_1, \ldots, v_{d_m}, \Lambda)$. In other words, $b(u, v_1, \ldots, v_{d_m}, \Lambda)$ is the number of different assignments of neighbors to other nodes given the assignments we made for u. Different choices of $u, v_1, \ldots, v_{d_m}$, and Λ result in different values of $b(u, v_1, \ldots, v_{d_m}, \Lambda)$, since each assignment of neighbors to u leaves slightly different degrees of freedom in the assignments of other nodes; e.g., if v is a neighbor of u in Λ, it can accommodate fewer additional assignments. Nevertheless, when n is large, the values of $b(\cdot)$ are similar, and for the sake of the analysis in this section we assume them to be equal. In section 6.2 we substantiate this assumption with a more accurate numerical computation and show that it has only a minor effect on our results.

In section 7.2 (Lemma 7.5) we show that under the assumptions of this section, the protocol is equally likely to reach each membership graph satisfying the sum degree invariant ($ds(u) = d_m$ for each u). Thus,

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

The only source of imprecision is the slight variation of the remaining degrees of freedom described above. Figure 6.1 compares these analytical results with a more precise numerical study (section 6.2). It shows 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 the same expectations.

6.2. Degree Markov chain. Allavena [1] analyzed the indegree distribution of a different protocol, with a constant outdegree, assuming no message loss, and using a one-dimensional MC. Since in S&F both node indegree and outdegree can vary, we construct a two-dimensional *degree MC*, where one dimension is indegree and the other is outdegree, reflecting their joint evolution at a single node.

A schematic diagram of the degree MC is shown in Figure 6.2. Recall that some actions where one of the selected view entries is empty have no effect on the views. We call such transformations *self-loop transformations* and do not show them in the diagram. 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,

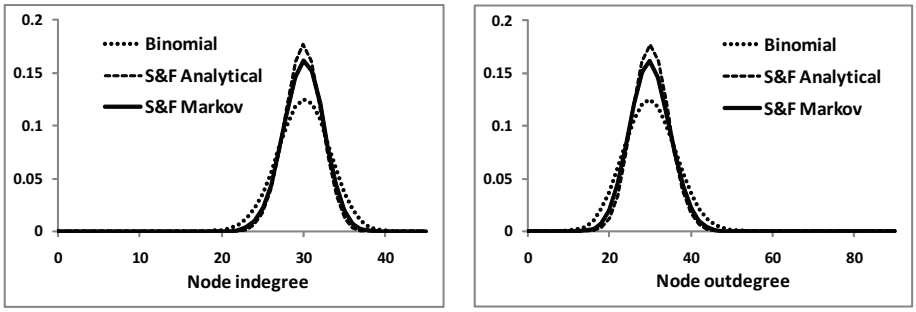


FIG. 6.1. *S&F node degree distributions (analytical approximation and exact, from degree MC) and binomial distributions with the same expectation.* $s = 90$, $d_{\mathrm{L}} = 0$, $\ell = 0$, $ds(u) = 90$ *for each* u *and arbitrary* $n \gg s$.

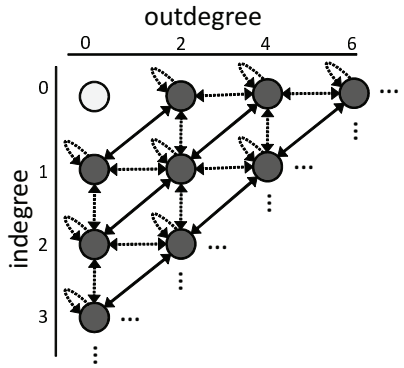


FIG. 6.2. *Degree MC. Dark circles are reachable states, and the light circle is an unreachable state. Solid lines correspond to (non-self-loop) transformations occurring with atomic actions (no loss, duplications, or deletions). Dashed lines correspond to (non-self-loop) transformations occurring due to loss, duplications, or deletions.*

when the loss is nonzero, $d_{\mathrm{L}} > 0$, so the outdegree cannot decrease to 0. With no loss, we allow $d_{\mathrm{L}} = 0$, but since the initial membership graph is weakly connected, by Lemma 6.2 no node can become isolated.

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 receiving 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 an 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 no point in computing probabilities for states with higher sum degrees. Therefore, for the sake of

the numerical computation we consider 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. This bound is used only to speed up numerical computation and is not used elsewhere. We verified that the bound does not affect our results by recomputing part of the results with higher bounds.

The resulting degree distributions, for $s = 90$, $d_{\mathrm{L}} = 0$, $\ell = 0$, and $ds(u) = 90$ for each u, are shown in Figure 6.1. Note that the figures show results from our analysis, which is independent of n, and hence the results hold for any $n \gg s$. We see that the degree distributions have a variance lower than that of the binomial distribution. It validates our analysis in section 6.1, which we use now to set the protocol's degree thresholds.

6.3. Setting the thresholds. We first select $\hat{d}$, the expected outdegree we are interested in, without loss. One should choose $\hat{d}$ based on the application needs 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 the expected outdegree is kept close to $\hat{d}$. Let δ be the maximum duplication and deletion probability that we are interested in. We then find d_{L} and s that satisfy, under no loss, the following conditions: (1) $\mathbb{E}(d(u)) = \hat{d}$, (2) $\Pr(d(u) \le d_{\mathrm{L}}) < \delta$, and (3) $\Pr(d(u) \ge s) < \delta$. For a given $\delta < 1/2$ we use (6.1) (where $d_m = 3\hat{d}$ by Lemma 6.3) to set

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

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

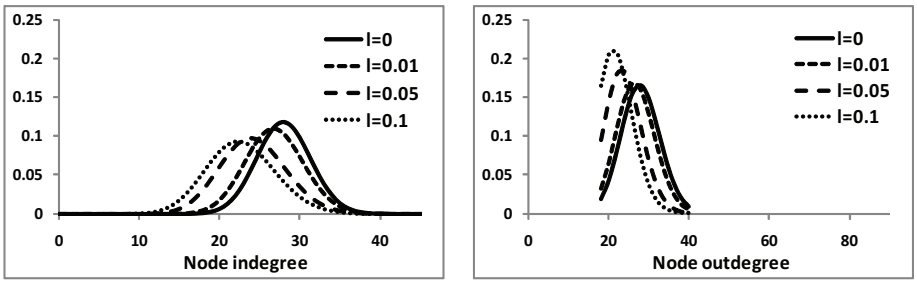
Since the values of d_{L} and s are discrete, $\Pr(d(u) \le d_{\mathrm{L}})$ and $\Pr(d(u) \ge s)$ are close but not necessarily equal. Consequently, the resulting expected outdegree may differ slightly from $\hat{d}$. For example, for $\hat{d} = 30$ and $\delta = 0.01$, d_{L} should be set to 18 and s to 40. 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 Property M1 (small views), as even constant-size (in the system size n) views are sufficient for the protocol to function properly.

6.4. Node degrees with loss. Figure 6.3 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. The average indegrees and their standard deviations are $28 \pm 3.4, 27 \pm 3.6, 24 \pm 4.1, 23 \pm 4.3$ for $\ell = 0, 0.01, 0.05, 0.1$, respectively.

It can be seen that while the average outdegree decreases with loss, it stays significantly above d_{L}, even for high loss rates. This could be counterintuitive as one might expect all outdegrees to eventually fall to d_{L}. However, due to the flexibility in node indegrees, even a slight decrease in the average outdegree triggers some duplications, thus preventing outdegrees from dropping to d_{L}. On the other hand, as we show in Lemma 6.7, the duplication probability is only slightly higher than the loss rate; i.e., duplications are not triggered more often than needed to compensate for lost ids. Therefore, even under relatively high loss rates, nodes are able to exchange ids effectively, without inducing excessive spatial dependencies.

Figure 6.3 shows that the indegree distribution remains concentrated around the expected degree. Thus, most nodes have similar indegrees, and we conclude that the



(a) (b)

FIG. 6.3. *S&F node degree distributions (exact, from degree MC) for different loss rates* $\ell = 0, 0.01, 0.05, 0.1$ *(*$d_{\mathrm{L}} = 18$*,* $s = 40$*) and arbitrary* $n \gg s$*.*

protocol satisfies Property M2 (load balance).

The next lemma proves what is evident from Figure 6.3—that the expected outdegree decreases with increasing loss (ℓ).

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 that the loss rate increases to $\ell_2 > \ell_1$. To accommodate a higher loss rate, the duplication probability has to increase to $dup_2 > dup_1$, while the deletion probability should not grow. For the duplication probability to increase, node outdegrees should reach their lower threshold d_{L} more frequently, and their upper threshold s at most as frequently as with ℓ_1. This, in turn, implies that the 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}. Hence, the variance of the node outdegree decreases (this can be observed in Figure 6.3(b)), and the following observation follows.

Observation 6.5. The deletion probability decreases with increasing ℓ.

This is illustrated in Figure 6.3(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.

We now characterize the connection between the probability of message loss and the probabilities of duplication and deletion performed by the protocol.

LEMMA 6.6. *In the steady state, the probability of duplication equals* ℓ *plus the probability of deletion.*

Proof. Since in the steady state the expected total number of edges remains constant, the number of new edges created by duplication equals the number of edges lost due to message loss or deletions. □

Recall (section 6.3) that δ is an upper bound on the duplication probability of the protocol with no loss. We get the following bound on duplications.

LEMMA 6.7. *In the steady state, the duplication probability during non-self-loop transformations is between* ℓ *and* $\ell + \delta$*.*

Proof. By Observation 6.5, for $\ell > 0$, the probability of deletion decreases below δ. By Lemma 6.6, the lemma follows. □

6.5. Degree dynamics of joining and leaving nodes. We now analyze how fast the membership graph is updated after a node joins or leaves (fails) in the steady state. That is, the system is in the steady state when a single join/leave happens. We

assume that a joining node starts with the minimal possible outdegree, d_{L}, and with indegree 0.

For a node u, an action initiated by u adds u's id to some view (unless the message is lost or the view is full), and an action whose target is u removes u's id from some view (unless a duplication is performed). Actions where u's id is sent from one node to another, on average, keep the same number of instances of u's id in the system, because in the steady state, the probability of duplication equals ℓ plus the probability of deletion.

Thus, there is an exponential decay of "old" instances of u's id in views (as a fixed percentage of these instances are chosen as message targets in every round), as well as a steady flow of "new" instances of u's id.

6.5.1. General lemmas. We first show that actions where u's id is sent from one node to another are expected to keep the same number of instances of u's id in the system.

LEMMA 6.8. *In the steady state, an action where v sends an instance of u's id to some w is expected to keep the number of instances of u's id unchanged.*

Proof. Consider an action where v sends an instance of u's id (in a message $[v, u]$) to w. There are four possible outcomes of this action (depicted in Figure 5.2). If the message is not lost and no duplication or deletion occurs, then the number of instances of u's id is unchanged. If the action performs a duplication and the message is lost or deleted, then views do not change at all.

The remaining two outcomes do change the number of id instances: (1) if the action performs a duplication, and the message is not lost or deleted, the number of instances of u's id increases by one; (2) if the action does not perform duplication but deletion or message loss occurs, we lose one instance of u's id. Note that the events of message loss and of deletion are mutually exclusive; i.e., the probability that both happen is 0. Denoting the probability of duplication by dup and the probability of deletion by del, the probability of (1) occurring is $dup(1 - (\ell + del))$, and the probability of (2) is $(1 - dup)(\ell + del)$. Since by Lemma 6.6, in the steady state $dup = \ell + del$, the probabilities of events (1) and (2) are equal. Therefore, in expectation, the number of instances of u's id is unchanged by actions that send it. ∎

For the sake of the following analysis we define a *round* to be the period of time during which each node is expected to initiate exactly one action.

The next lemma bounds the rate at which instances of u's id disappear from views of other nodes. We start from some round t_0. Note that although new instances of u's id may be added during the period we analyze, we consider only old instances that were created up to round t_0.

LEMMA 6.9. *Consider round t_0 in the steady state. The probability that an instance of u's id remains in the system from round t_0 to round $t_0 + i$ is bounded from above by*

$$\left(1 - \frac{(1 - \ell - \delta)\, d_{\mathrm{L}}}{s^2}\right)^i.$$

Proof. By Lemma 6.8, only actions where u is the target of a message change the expected number of old instances of u's id in the system. Let v be a node that has u in its view, and suppose that v initiates an action. The id of u is deleted from v's view as a result of the following sequence of events: (1) v selects two nonempty entries in its view (this happens with probability $(\frac{d(v)}{s})^2$); (2) the first selected entry

(message target) contains u's id (this happens with probability $\frac{1}{d(v)}$ given (1)); and (3) the action does not perform duplication (this happens with probability *dup* given (1) and (2); we analyze *dup* later). Then, the probability that the id of u is removed from v's view is

$$\left(\frac{d(v)}{s}\right)^2 \cdot \frac{1}{d(v)} \cdot (1 - dup) \;=\; \frac{(1 - dup)\, d(v)}{s^2}.$$

Note that *dup* is not equal to the system-wide average duplication probability since we are considering only nodes that have instances of u's id in their view, thus preferring nodes with higher outdegrees. Fortunately, since the duplication probability decreases with an increasing node outdegree, *dup* is lower than the system-wide duplication probability. Thus, we use Lemma 6.7 to bound *dup* by the system-wide upper bound on the duplication probability $\ell + \delta$, getting

$$\frac{(1 - dup)\, d(v)}{s^2} \;\geq\; \frac{(1 - \ell - \delta)\, d(v)}{s^2}.$$

Finally, we use the fact that $d(v) \leq d_{\mathrm{L}}$ and obtain the following lower bound on the probability of removal of each instance of u's id in the system during a single round:

$$\frac{(1 - \ell - \delta)\, d(v)}{s^2} \;\geq\; \frac{(1 - \ell - \delta)\, d_{\mathrm{L}}}{s^2}.$$

Since all the events during a round happen independently of other rounds, at the end of round $t_0 + i$, the probability that an instance of u's id remains in the system from time t_0 is at most

$$\left(1 - \frac{(1 - \ell - \delta)\, d_{\mathrm{L}}}{s^2}\right)^i. \qquad \square$$

6.5.2. Representation of leaving nodes. The following lemma follows directly from Lemma 6.9.

LEMMA 6.10. *Consider node u leaving (or failing) at round t_0 when the system is in the steady state and an instance of u's id in some other node's view. Then the probability for this instance to still be in some view at round $t_0 + i$ is bounded from above by*

$$\left(1 - \frac{(1 - \ell - \delta)\, d_{\mathrm{L}}}{s^2}\right)^i.$$

Figure 6.4 illustrates the result of Lemma 6.10. It shows the evolution of the upper bound on the probability of an id instance remaining in the system for several different loss rates and the values $d_{\mathrm{L}} = 18$ and $s = 40$ as in the examples in previous sections. It demonstrates that (the bound on) the id instance decay rate is almost unaffected by loss, and that after merely 70 rounds (i.e., after each node initiates about 70 actions), fewer than 50% of the id instances of a left/failed node are expected to remain in the system.

6.5.3. Representation of joining nodes. Let the expected indegree of a node (in the steady state, under a uniform distribution over the nodes) be D_{in}. We denote

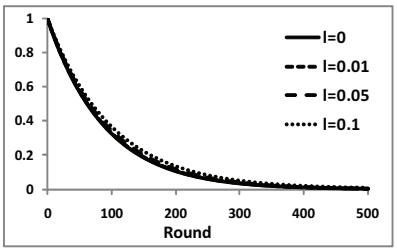


FIG. 6.4. *The upper bound on the probability that an id instance of a left/failed node remains in the system as a function of time since the leave/failure, for loss rates* $\ell = 0, 0.01, 0.05, 0.1$ *(*$\delta = 0.01$*,* $d_{\mathrm{L}} = 18$*,* $s = 40$*) and arbitrary* $n \gg s$*.*

by Δ the *expected creation rate*—the expected number of new id instances created by an average node u during a round. We bound Δ in the following lemma.

LEMMA 6.11. *In the steady state,*

$$\Delta \;\geq\; \frac{(1-\ell-\delta)\, d_{\mathrm{L}}}{s^2} \cdot D_{\mathrm{in}}.$$

Proof. Clearly, in the steady state, to compensate for the decaying id instances, u creates the same number of new id instances in expectation. From Lemma 6.9, the expected number of the instances of u's id that are removed from views during a round is at least $\frac{(1-\ell-\delta)\, d_{\mathrm{L}}}{s^2} \cdot d_{\mathrm{in}}(u)$. Taking the expectation,

$$\begin{aligned}\Delta &\geq \mathbb{E}\left(\frac{(1-\ell-\delta)\, d_{\mathrm{L}}}{s^2} \cdot d_{\mathrm{in}}(u)\right)\\ &= \frac{(1-\ell-\delta)\, d_{\mathrm{L}}}{s^2} \cdot \mathbb{E}(d_{\mathrm{in}}(u)) \;=\; \frac{(1-\ell-\delta)\, d_{\mathrm{L}}}{s^2} \cdot D_{\mathrm{in}}. \qquad \square\end{aligned}$$

LEMMA 6.12. *If a new node joins when the systems is in the steady state, the expected creation rate of the newly joined node is at least*

$$\left(\frac{d_{\mathrm{L}}}{s}\right)^2 \cdot \Delta.$$

Proof. A new instance of u's id can be added to some view only as a result of a non-self-loop action initiated by u. The probability of such a non-self-loop action is $(\frac{d(v)}{s})^2$. For a veteran node in the system, this probability may be as high as $(\frac{s}{s})^2$. For a newly joined node, this probability may be as low as $(\frac{d_{\mathrm{L}}}{s})^2$. The lemma follows from Lemma 6.11 and the ratio of the above probabilities: $(\frac{d_{\mathrm{L}}}{s})^2/(\frac{s}{s})^2 \;\geq\; (\frac{d_{\mathrm{L}}}{s})^2$. $\square$

LEMMA 6.13. *If a new node joins when the systems is in the steady state, during its first* $\frac{s^2}{(1-\ell-\delta)\, d_{\mathrm{L}}}$ *rounds the node is expected to create at least* $(\frac{d_{\mathrm{L}}}{s})^2 \cdot D_{\mathrm{in}}$ *instances of its id in other views.*

Proof. By Lemma 6.11, a veteran node is expected to create at least D_{in} new instances of its id in at most $\frac{s^2}{(1-\ell-\delta)\, d_{\mathrm{L}}}$ rounds. By Lemma 6.12, the expected creation rate of the newly joined node is at most $(\frac{d_{\mathrm{L}}}{s})^2$ times slower. Thus, during the same number of rounds, the newly joined node is expected to create at least $(\frac{d_{\mathrm{L}}}{s})^2 \cdot D_{\mathrm{in}}$ instances of its id in other views. $\square$

The above result may be hard to parse, so we substitute some typical values to obtain a more intuitive result in the following corollary.

COROLLARY 6.14. *For $\ell+\delta \ll 1$ and $s/\,d_{\mathrm{L}} = 2$, after $2s$ rounds, a newly joined node is expected to create at least $D_{\mathrm{in}}/4$ instances of its id in other views.*

Note that after creating $D_{\mathrm{in}}/4$ new in-neighbors, the new node is likely to receive messages from these neighbors, thus increasing its outdegree to above d_{L} and making the duplication probability at the node low. We conclude that under moderate loss, after roughly $2s$ rounds, the new node can efficiently engage in the protocol and becomes integrated in the system.

7. Uniformity and independence. In this section we analyze the remaining protocol properties of uniformity and independence (Properties M3–M5). In section 7.1 we define a global MC 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 a weakly connected 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 a view of any other node in the system. 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.

Since in this section we are interested in the steady state behavior of the protocol, we assume that the churn ceases for the period we analyze. We further assume that the initial topology (i.e., the one reached after the churn stops) satisfies some minimal connectivity conditions (formally specified below). In practice, such conditions will be satisfied if the churn is moderate. If the churn is severe enough to partition the network, not only is our analysis not applicable, but also no gossip-based protocol can be expected to work well. In section 6.5 we analyze the time it takes to integrate new nodes and to remove id instances of left/failed ones.

7.1. The global Markov chain graph. We define $\mathcal{G}(s, d_{\mathrm{L}}, \ell)$ to be the *global MC graph* induced by S&F with given s, d_{L}, and ℓ. For simplicity, we omit the parameters and refer to this graph as $\mathcal{G}$. We call vertices in $\mathcal{G}$ *states*, as each vertex represents a global state of the views of all nodes. The set of vertices of $\mathcal{G}$ can be represented as a union $\mathcal{V} = \mathcal{V}_0 \bigcup \mathcal{V}_1$ of two disjoint sets of states: $\mathcal{V}_0$ that contains all weakly connected membership graphs, where all node outdegrees are between d_{L} and $s-2$ (inclusive) and are even; and $\mathcal{V}_1$ that contains all weakly connected membership graphs that are not in $\mathcal{V}_0$ (i.e., membership graphs where some nodes have outdegree of s) and that can be reached by S&F transformations from some membership graph in $\mathcal{V}_0$. 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 the probabilities of all transformations from G_1 to G_2.

Note that some membership graphs are partitioned, e.g., when some node has no incoming edges and all of its outgoing edges are self-edges. Since partitioned states are excluded from $\mathcal{G}$, we replace the edges leading to them from states in $\mathcal{G}$ 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.

We also exclude states that are unreachable from the largest connected component of $\mathcal{G}$. Such unreachable states are (some of the) membership graphs where some nodes have full views, i.e., outdegrees of s. Nodes with full views cannot effectively exchange ids with their neighbors (which may also have full views). For example, states where all views are full are clearly unreachable by S&F transformations. In the analysis we

assume that the system begins from a reachable state; i.e., the initial state is in $\mathcal{G}$ and not among the unreachable states.

Note that each state in $\mathcal{G}$ has a self-loop edge corresponding to self-loop transformations, which 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 the appendix.

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 MC 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 MC is ergodic and, by the fundamental theorem of the theory of MCs, 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, $0 < ds(u) \leq s$ 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 a vector mapping each node to its sum degree. For the sake of the analysis 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 $\bar{\mathbf{ds}}$ is the sum degree vector of the initial state.

We now prove 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 [27] to S&F. We first observe that $\mathcal{G}_{\bar{\mathbf{ds}}}$ is in fact undirected.

LEMMA 7.3. *$\mathcal{G}_{\bar{\boldsymbol{ds}}}$ is reversible.*

Proof. Consider an arbitrary $G \in \mathcal{G}_{\bar{\mathbf{ds}}}$, an arbitrary transformation initiated by node u, sending u and w to v, and the resulting $G' \in \mathcal{G}_{\bar{\mathbf{ds}}}$. Clearly, G' can be transformed back into G by v sending v and w to u. By Proposition 5.2, all transitions happen with the same probability. The lemma follows. ∎

LEMMA 7.4. *The outdegrees and the indegrees of all states in $\mathcal{G}_{\bar{\boldsymbol{ds}}}$ are equal.*

Proof. G's outdegree is the sum of the probabilities of all transformation of G. Since each transformation involves an arbitrary node and by Proposition 5.2, the probability of each transformation is the same. ∎

By Lemmas 7.4 and 7.3, $\mathcal{G}_{\bar{\mathbf{ds}}}$ induces a doubly stochastic MC transition matrix.

LEMMA 7.5. *The stationary distribution of the MC on $\mathcal{G}_{\bar{\boldsymbol{ds}}}$ is the uniform distribution over all states in $\mathcal{G}_{\bar{\boldsymbol{ds}}}$.*

Proof. Consider the MC induced by $\mathcal{G}_{\bar{\mathbf{ds}}}$. Clearly, $\mathcal{G}_{\bar{\mathbf{ds}}}$ is finite. From Lemma A.2 it is irreducible. It is aperiodic (meaning that the greatest common denominator of

the lengths of directed paths connecting any two nodes in $\mathcal{G}_{\bar{\mathbf{ds}}}$ is 1) since each state $G \in \mathcal{G}_{\bar{\mathbf{ds}}}$ has a self-edge. From the above, the MC is ergodic and, by the fundamental theorem of the theory of MCs, has a unique stationary distribution.

By Lemma 7.3, $\mathcal{G}_{\bar{\mathbf{ds}}}$ is undirected. On undirected graphs, the probability of each state under the stationary distribution is proportional to its degree. Since by Lemma 7.4 the degrees of all states are equal, the stationary distribution of an MC on graph $\mathcal{G}_{\bar{\mathbf{ds}}}$ is uniform. □

7.3. Proving uniformity (Property 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 of containing any other id $v \neq u$.

LEMMA 7.6. *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 of appearing in u's view. □

7.4. Proving spatial independence (Property 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. This assumption allows us to show (in Lemma 7.9) that under moderate loss, α converges to a much higher value that depends on the actual loss. Thus, α remains higher than 2/3.

Assumption 7.7. Initially, $\alpha \geq 2/3$.

Note that due to Assumption 7.7 our analysis is not applicable for high loss rates or high churn rates when all new joiners start with the same initial view, making α too low. Nevertheless, since our analysis is not tight, we speculate that the protocol may also work well 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. By Lemma 6.7, duplication probability is at most $\ell + \delta$ (recall that δ is an upper bound on the duplication probability of the protocol with no loss).

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 [32, 4]) and δ (see section 6) are on the order of 1%; hence the vast majority of view entries are expected to be independent.

The following lemma coarsely bounds the probability of a dependent view entry that u sends returning 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 Lemma 7.9.

LEMMA 7.8. *Suppose u sends a dependent entry to one of its neighbors. In the steady state, the probability of this entry being 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.

Proof. We (crudely) bound the probability of a dependent entry being sent back to its originator as follows. In the worst case, when all dependent entries of u's out-neighbors point to u, the probability of u getting back a dependent entry from its immediate neighbor is at most $1-\alpha(1-1/n)$. For simplicity, we neglect $1/n$ (assuming $n \gg 1$) and thus use $1-\alpha$ for the above bound. More generally, the probability of a dependent entry getting back to u after traversing i edges under the worst case assumptions that all dependent entries of all nodes reachable from it by i edges are "devoted" to such back edges to u is bounded by $(1-\alpha)^i$. Thus, the probability of a given dependent entry returning to u after being removed from u's view is bounded by

$$\sum_{i=1}^{\infty}(1-\alpha)^i \;=\; \frac{1}{1-(1-\alpha)} - 1 \;=\; \frac{1}{\alpha} - 1.$$

Since we assumed that $\alpha \geq 2/3$ (Assumption 7.7), the above expression is at most $1/2$. ∎

Note that the above bound is not tight due to the following worst-case assumptions: (1) for each $i \in [1,\infty)$, all dependent entries of all nodes reachable from u by i edges are devoted to edges to u; (2) we ignore the probability of the entry disappearing due to loss or deletions; and (3) summing the return probabilities for all i, we ignore the fact that if the entry returns after traversing i edges, it will not return after traversing j edges for $j > i$.

LEMMA 7.9. *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 MC* to model the state of the entry, which can be either "dependent" or "independent."

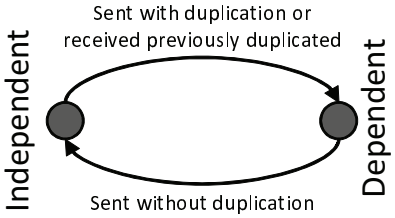


FIG. 7.1. *Dependence MC.*

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 7.1, 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 the probability of going from the independent to the dependent state. By Proposition 5.2 each entry has the same probability of being

involved in a transformation. Thus, by Lemma 6.7, the probability of an entry becoming dependent during a non-self-loop transformation is at most $\ell+\delta$. By Lemma 7.8, the probability of getting back a dependent entry given that it was duplicated at the time of sending 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 6.7, the probability of (2) is at least $1-(\ell+\delta)$. We next bound the probability of (1).

Let β be the probability of an entry being a *self-edge*, i.e., $u.lv[i] = u$. The most likely scenario for creating a self-edge in u's view is that (1) u creates two parallel edges (v,u) by initiating two actions involving one of its out-neighbors, 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), sending a message $[v,u]$ to u that is not lost or deleted. Since the probability of (2) is at most 1/2 by Lemma 7.8, we conclude that at most half of the dependent entries are self-edges. Since we assumed $\alpha \geq 2/3$ (Assumption 7.7), the probability β of a random view entry being 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:

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

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 [15]. Although we do not know the exact distribution of the number of independent ids in views, since the loss rate (and hence the duplication probabilities) is uniform and independent, we speculate that the number of independent ids in node views is distributed similarly to node outdegrees but with lower expectation ($\alpha\, d_{\mathrm{E}}$ instead of d_{E}). That is, the number of independent ids in a view is distributed so that it is 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 having fewer than three independent neighbors is at most ϵ. For example, for $\ell = \delta = 1\%$ and $\epsilon = 10^{-30}$, d_{L} should be set to at least 26.

7.5. Proving temporal independence (Property M5). We next analyze Property 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 we are interested in convergence time from an *average* state $\tilde{G}$ distributed according to π, and not from an *arbitrary* state (the latter is called mixing time). This is because such a worst-case assumption inevitably yields overly pessimistic bounds that do not shed much light on the protocol's behavior in practice. Indeed, mixing time analyses of similar MCs in previous works [14, 10, 11] proved bounds on the order of $O(n^9)$ steps or more, which can hardly be considered useful in practice. We instead start from an average state, which provides meaningful bounds, albeit for more limited circumstances. In particular, if the churn rate is high and all new joiners start with the same initial view, convergence might be slower.

For the sake of this analysis, we assume that there are exactly n nodes, fixed during the period we analyze, in all states in $\mathcal{G}$ and that $s \ll \sqrt{n}$. We first derive 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.

Our analysis makes use of the following well-established notions of neighbor set and boundary.

Definition 7.10 (neighbor set). *Let x be a vertex in $\mathcal{G}$. Then, the* neighbor set *of x, $\Gamma_i(x)$ is the subset of $\mathcal{V}$ reachable from x by paths of at most i edges.*

Recall (section 3.2) that $P(x, y)$ is the transition probability of the MC from state x to y. Intuitively, the boundary size of S is the "flow" from S to the rest of the graph relative to the stationary distribution π.

Definition 7.11 (boundary size). *For $x, y \in \mathcal{V}$, let $Q(x, y) = \pi(x)P(x, y)$, and for $A, B \subset \mathcal{V}$, let $Q(A, B) = \sum_{x\in A, y\in B} Q(x, y)$. The* boundary size *of $S \subset \mathcal{V}$, $|\partial S|$, is then $|\partial S| = Q(S, S^c)$, where $S^c = \mathcal{V} \setminus S$ is the complement of S.*

Definition 7.12 (conductance). *The* conductance *of $S \subset \mathcal{V}$, $\phi(S)$, is defined as follows: $\phi(S) = \frac{|\partial S|}{\pi(S)}$. The conductance of graph $\mathcal{G}$ is defined as follows: $\phi(\mathcal{G}) = \min_{S\subset\mathcal{V}:\pi(S)\le 1/2}(\phi(S))$.*

As explained above, we focus on starting from a random state rather than from an arbitrary one. We thus introduce the new notion of *expected conductance*.

Definition 7.13 (expected conductance). *The expected conductance of graph $\mathcal{G}$, $\Phi(\mathcal{G})$, is defined as follows:*

$$\Phi(\mathcal{G}) \;=\; \mathbb{E}\left(\min_{i:\pi(\Gamma_i(X))\le 1/2}(\phi(\Gamma_i(X)))\right),$$

where X is a random state in $\mathcal{V}$ distributed according to π.

The following lemma bounds the expected conductance of $\mathcal{G}$.

Lemma 7.14. *Assume $s \ll \sqrt{n}$. Then, the expected conductance of $\mathcal{G}$ satisfies $\Phi(\mathcal{G}) \ge \frac{d_{\mathrm{E}}(d_{\mathrm{E}}-1)\,\alpha}{2\,s(s-1)}$.*

Proof. Recall the definition of the expected conductance:

$$\Phi(\mathcal{G}) \;=\; \mathbb{E}\left(\min_{i:\pi(\Gamma_i(X))\le 1/2}(\phi(\Gamma_i(X)))\right),$$

where X is distributed according to π, and

$$\phi(\Gamma_i(X)) = \frac{\sum_{x\in\Gamma_i(X)}(\pi(x)\sum_{y\in\Gamma_i(X)^c}P(x,y))}{\pi(\Gamma_i(X))}.$$

We bound $\sum_{y\in\Gamma_i(X)^c}P(x,y)$—the sum of all transition probabilities from x to states in $\Gamma_i(X)^c$—as follows: Recall that each two entries in a view of each node have the same probability of being involved in a transformation. We thus have $n\cdot s\cdot(s-1)$ view entry pairs in x, each involved in a transformation with probability $\frac{1}{n\cdot s\cdot(s-1)}$. We now bound the probability of a random transformation from a random state in $\Gamma_i(X)$ leading to one of the states in $\Gamma_i(X)^c$. The probability of both view entries being nonempty is $\frac{d_{\mathrm{E}}(d_{\mathrm{E}}-1)}{s(s-1)}$, and the probability of each of them pointing to a random node independently of other view entries is α. Thus, a random transformation has the probability of at least $\frac{d_{\mathrm{E}}(d_{\mathrm{E}}-1)\,\alpha}{s(s-1)}$ of leading to one of the states in $\Gamma_i(X)^c$, independently of other transformations. Due to the assumption that $s \ll \sqrt{n}$, the probability of several such independent transformations leading to the same state in $\Gamma_i(X)^c$ is negligible for small $\Gamma_i(X)$ and is at most half when $\pi(\Gamma_i(X)) \approx 1/2$. (More frequent duplicate selections would imply that there is a higher fraction than $1-\alpha$ of dependent entries, since duplicate selection is caused by several different sequences of transformation reaching the same state.) Thus,

$$\Phi(\mathcal{G}) \;\ge\; \frac{d_{\mathrm{E}}(d_{\mathrm{E}}-1)\,\alpha}{2\,s(s-1)}. \qquad \square$$

We now use standard techniques typically used to deduce the mixing time from conductance to show the following lemma.

LEMMA 7.15. *Assuming* $s \ll \sqrt{n}$,

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

Proof. The MC mixing time $T_\epsilon(\mathcal{G})$ is related to the MC graph conductance as follows [30]:

$$T_\epsilon(\mathcal{G}) \;\le\; 1+\frac{4}{\phi^2(\mathcal{G})}\left(\log\frac{1}{\pi_*}+\log\frac{4}{\epsilon}\right),$$

where $\pi_* = \min_{x\in\mathcal{V}}\pi(x)$ is the probability, under stationary distribution, of a least probable "worst-case" state. Since we are starting from a random state X distributed according to π, we use $\Phi(\mathcal{G})$ instead of $\phi(\mathcal{G})$, and $\pi' = \mathbb{E}(\pi(X))$ instead of π_*. Thus,

$$\tau_\epsilon(\mathcal{G}) \;\le\; 1+\frac{4}{\Phi^2(\mathcal{G})}\left(\log\frac{1}{\pi'}+\log\frac{4}{\epsilon}\right).$$

As we do not know the distribution π explicitly, we bound $\mathbb{E}(\pi(X))$ from below as if each state had the same probability. In each state in $\mathcal{G}$, each node selects, uniformly

at random, at most s neighbors out of n nodes independently of other selections. Thus, there are at most $n^{n\,s}$ different states in $\mathcal{G}$. Since some states have higher probability relative to π than the others (e.g., since most views are expected to contain fewer than s entries),

$$\mathbb{E}(\pi(X)) \;\geq\; \frac{1}{n^{n\,s}}.$$

Substituting the result of Lemma 7.14, we get

$$\begin{aligned}\tau_\epsilon(\mathcal{G}) &\leq \frac{16\,s^2(s-1)^2}{{d_{\mathrm{E}}}^2(d_{\mathrm{E}}-1)^2\,\alpha^2}\left(\log(n^{n\,s}) + \log\frac{4}{\epsilon}\right)\\ &= \frac{16\,s^2(s-1)^2}{{d_{\mathrm{E}}}^2(d_{\mathrm{E}}-1)^2\,\alpha^2}\left(n\,s\cdot\log(n) + \log\frac{4}{\epsilon}\right). \qquad \square\end{aligned}$$

Note that for zero loss and $\alpha = 1$, 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 nonatomic 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.

Appendix. Uniformity and independence. In this appendix we show that the global MC graph is strongly connected. We first prove this in Lemma A.2 for the loss-free case, and then prove the general case with positive loss in Lemma 7.1. Recall that the sum degree of node u, $ds(u)$, is equal to $d(u) + 2\,d_{\mathrm{in}}(u)$. In the loss-free case, the sum degrees remain invariant. We define the following loss-free transformations on membership graphs.

Edge exchange transformation of (u, w) *and* (v, z). This transformation removes edges (u, w) and (v, z) and creates edges (u, z) and (v, w) instead. First, assume that u and v are connected by an edge (u, v). A prerequisite for this transformation is that $d(u) > d_{\mathrm{L}}$ and $d(v) < s$. We use this transformation only when the prerequisite holds. The following two S&F actions implement the edge exchange transformation: u initiates an action, selects entries containing v and w in its view, removes these entries from its view, and sends a message $[u, w]$ to v. On receiving the message, v creates an edge (v, u). Then, v initiates an action and sends $[v, z]$ to u (note that v necessarily has u in its view), and u creates an edge (u, z). It is easy to see that except for the edge exchange, the rest of the membership graph remains unchanged.

We now generalize the edge exchange to any two nodes u and v that are not necessarily neighbors. Since the graph is weakly connected, there exists at least one undirected path between u and v. Let this path be $u, y_1, y_2, \ldots, y_k, v$. We use simple

edge exchange between neighbors to "send" the edges we want to exchange along the path. That is, u exchanges edge (u, w) with some arbitrary y_1's edge, say (y_1, x_1). Then, y_1 exchanges edge (y_1, w) with y_2 and so on, until y_k exchanges edge (y_k, w) with v's edge (v, z). Now y_k exchanges edge (y_k, z) with y_{k-1}'s edge (y_{k-1}, x_k). This way, an edge to z travels towards u while returning the temporarily misplaced edges $x_1, x_2, \ldots, x_k$ to their original owners. A prerequisite for the generalized edge exchange transformation between u and v is the existence of an undirected path between u and v such that, for each two neighbors in the path connected by an edge (y_1, y_2), $d(y_1) > d_{\mathrm{L}}$ and $d(y_2) < s$.

Degree borrowing transformation between u and v. The goal of this transformation is to decrease the outdegree of node u, and to increase the outdegree of node v, while keeping their sum degrees invariant. We first define a degree borrowing transformation between two neighbor nodes u and v and later generalize it to two arbitrary nodes. Obviously, a prerequisite for this transformation is that $d(u) > d_{\mathrm{L}}$ and $d(v) < s$. Degree borrowing is then implemented by u initiating an action and sending a message to v that is not lost.

Degree borrowing between two arbitrary nodes u and v is then implemented as follows: We identify another node w, such that there exists an edge (w, v), and exchange an arbitrary u's edge (u, z) with w's edge (w, v), thus making u and v neighbors. We then proceed with degree borrowing between neighbors. A prerequisite for the generalized degree borrowing transformation between u and v is a nonzero indegree of v and the ability to perform edge exchange between u and at least one of v's in-neighbors.

Recall (section 7.2) that $\bar{\mathbf{ds}} = (ds(u_1), ds(u_2), \ldots)$ is a vector mapping each node to its sum degree, and that $\mathcal{G}_{\bar{\mathbf{ds}}}$ is the subgraph of $\mathcal{G}$ where in all states all node sum degrees are according to $\bar{\mathbf{ds}}$. The next lemma proves that, in a static setting with n nodes and when each pair of nodes satisfies the prerequisite for edge exchange, $\mathcal{G}_{\bar{\mathbf{ds}}}$ is strongly connected.

LEMMA A.1. *When in each state in $\mathcal{G}_{\bar{ds}}$, each two nodes satisfy the prerequisite for edge exchange, $\mathcal{G}_{\bar{ds}}$ is strongly connected.*

Proof. We show that, for each $G, G' \in \mathcal{G}_{\bar{\mathbf{ds}}}$, there exists a sequence of transformations transforming G to G'. We use only transformations not involving loss, duplications, or deletions. We transform G into G' in two steps: (1) Transform G into G^* such that node outdegrees in G^* are equal to those in G' (note that by the sum degree invariant, the indegrees become equal, too); and (2) transform G^* into G'.

We implement (1) as follows: We iteratively identify pairs of nodes so that one has outdegree higher than its outdegree in G' and another has outdegree lower than its outdegree in G'. Since the total number of edges in the membership graph remains constant, such pairs are guaranteed to exist as long as at least one node has an outdegree different from its outdegree in G'. For each such pair, we invoke the degree borrowing transformation, making the outdegrees of the two nodes closer to their outdegrees in G'. Note that since degree borrowing does not alter node sum degrees, as a result of the transformation we get a state that is in $\mathcal{G}_{\bar{\mathbf{ds}}}$. Clearly, after a finite number of such transformations, we get G^*, where node outdegrees are equal to those in G'.

To implement (2) we repeatedly identify "misplaced" edges and use edge exchange transformations to move them to the nodes to which they belong according to G'. As the number of edges in the membership graph is finite, a finite number of such transformations is needed to transform G^* into G'. ∎

The next lemma proves that, with no loss (i.e., $\ell = 0$ and $d_{\mathrm{L}} = 0$) and for $\bar{\mathbf{ds}}$ such that for each u, $0 < ds(u) \leq s$, $\mathcal{G}_{\bar{\mathbf{ds}}}$ is strongly connected.

LEMMA A.2. *When $0 < ds(u) \leq s$ for each u, $\ell = 0$, and $d_{\mathrm{L}} = 0$, $\mathcal{G}_{\bar{ds}}$ is strongly connected.*

Proof. We first show that in the setting of the lemma, in each state in $\mathcal{G}_{\bar{\mathbf{ds}}}$, each two nodes that do not satisfy the prerequisite for edge exchange (outdegree above d_{L} for the initiating node and outdegree below s for the other node) can temporarily increase/decrease their outdegree using degree borrowing with one of their neighbors. The lemma then follows from Lemma A.1.

Since $0 < ds(u) \leq s$ for each u and $d_{\mathrm{L}} = 0$, if $d(u) = d_{\mathrm{L}} = 0$, then, by the sum degree invariant, u has at least one in-neighbor y such that $0 < d(y)$. Similarly, if $d(u) = s$, then u has at least one out-neighbor y such that $d(v) < s$. Thus, for any node that has an outdegree of d_{L} or s, we can perform degree borrowing before and after the edge exchange so that the node satisfies the edge exchange prerequisite. The degree borrowing performed after the edge exchange involves the same nodes as the one performed before the edge exchange, thus eliminating any effects of degree borrowing on the membership graph. We also do this for edge exchange transformations used within degree borrowing. Thus, for every u whose outdegree is lower than $s-2$ and every v whose outdegree is greater than 2, the prerequisites for degree borrowing of u from v can be satisfied. □

We now take message loss into account ($\ell > 0$) and show that $\mathcal{G}$ is also strongly connected. Recall (section 7.1) that the states of $\mathcal{G}$ include states in $\mathcal{V}_0$ (where all node outdegrees are between d_{L} and $s-2$ and are even) and states in $\mathcal{V}_1$ (where some nodes have outdegrees of s) that are reachable by S&F from $\mathcal{V}_0$.

LEMMA 7.1 (restated). *When $0 < \ell < 1$, $\mathcal{G}$ is strongly connected.*

Proof. We prove the lemma in several steps. We first prove that any two states in $\mathcal{V}_0$ are reachable from each other (Lemma A.3), and then show that there is a path from any state in $\mathcal{V}_1$ to some state in $\mathcal{V}_0$ (Lemma A.4). In the following two lemmas, unless specified otherwise, we consider transformations that do not involve message loss.

LEMMA A.3. *For each $G, G' \in \mathcal{V}_0$, there exists a sequence of S&F transformations transforming G to G'.*

Proof. We first construct from G' another membership graph G'' by adding outgoing edges from every node whose outdegree in G' is d_{L} to two arbitrary nodes. Note that G'' is also in $\mathcal{V}_0$. Clearly, G'' can be transformed to G' by invoking S&F transformations involving only these additional edges, where these edges are lost. The remainder of the proof is dedicated to transforming G to G''. Note that since in section 5 we require $d_{\mathrm{L}} \leq s-6$, we are guaranteed that $s-2 > d_{\mathrm{L}}+2$.

We start by transforming G into G_1 where each node has outdegree of at least $d_{\mathrm{L}}+2$. We first increase the outdegrees of nodes with outdegree d_{L}. We pick u such that $d(u) = d_{\mathrm{L}}$ and perform the following transformation: If u has an in-neighbor with outdegree of at least $d_{\mathrm{L}}+4$, we invoke an S&F transformation where this neighbor sends a message to u, thus increasing its outdegree to $d_{\mathrm{L}}+2$. If u does not have an in-neighbor with outdegree of at least $d_{\mathrm{L}}+4$, we invoke an S&F transformation where u sends a message to any of its out-neighbors (involving duplication), and then a transformation where that neighbor sends a message back to u. Thus, the outdegree of u becomes $d_{\mathrm{L}}+2$, while other node outdegrees do not change.

From now on, we maintain the outdegrees of all nodes in the range $[d_{\mathrm{L}}+2, s-2]$. Thus, the prerequisites for edge exchange and degree borrowing transformations between any two nodes are satisfied.

We next transform G_1 into G_2, where the total number of edges is as in G''. To decrease the number of edges, we invoke S&F transformations involving loss at nodes whose outdegree is still greater than $d_L + 2$. To increase the number of edges, we need to invoke S&F transformations that perform duplication, which happens only when a node has outdegree of d_L. To this end, we pick an arbitrary node u and perform degree borrowing transformations to decrease the outdegrees of u and of all of its out-neighbors to d_L. Once u reaches an outdegree of d_L, we invoke S&F transformations where u sends messages to its out-neighbors and performs duplications until the neighbors' outdegrees reach $s-2$ (or the desired number of edges is reached). We then invoke S&F transformation where one of u's in-neighbors sends u a message, thus increasing u's outdegree to $d_L + 2$. We continue the above process (possibly repeating it with different nodes), until we reach the desired number of edges. All subsequent transformations will preserve the total number of edges in the membership graph.

We next transform G_2 into G_3, where for each node u, its sum degree is as in G''. We iteratively identify pairs of nodes u and v so that $ds(u)$ is too low and $ds(v)$ is too high until for each u, $ds(u)$ is as in G''. (Such pairs are guaranteed to exist as long as at least one node u has a sum degree different from that in G''.) For such a pair u, v, we identify an arbitrary node w and use edge exchanges between w and the in-neighbors of u and v to create edges (w, u) and (w, v). (If u or v do not have in-neighbors, we perform degree borrowing to create the needed in-neighbors.) We then temporarily decrease the outdegree of w to d_L using degree borrowing (as described earlier), and perform the following sequence of S&F transformations between w and its arbitrary out-neighbor $y \neq u, v$: (1) w sends and duplicates $[w, u]$ to y, thus creating edges (y, w) and (y, u); (2) y sends $[y, u]$ to w, removing edges (y, w) and (y, u) and creating edges (w, y) and (w, u) (both these edges now have multiplicity of at least 2); (3) w sends $[w, v]$ to y, and the message is lost, thus removing edges (w, y) and (w, v). The outcome of this entire sequence is creating one new incoming edge to u and removing one incoming edge from v, thus increasing u's sum degree by 2 and decreasing v's sum degree by 2. The total number of edges in the membership graph remains unchanged. We now can undo all degree borrowing transformations so that node outdegrees are again between $d_L + 2$ and $s - 2$. After a finite number of such transformations, we get G_3, where for each node u, $ds(u)$ is as in G''.

By Lemma A.1, G'' is reachable from G_3, and the lemma follows. □

We next prove that there is a path from any state in $\mathcal{V}_1$ to some state in $\mathcal{V}_0$.

LEMMA A.4. *For each $G \in \mathcal{V}_1$, there exists a sequence of transformations transforming G to some $G' \in \mathcal{V}_0$.*

Proof. In order to get from G to some $G' \in \mathcal{V}_0$ we need to decrease the outdegrees of all nodes to at most $s-2$. To this end, we iteratively pick nodes having outdegrees of s, and initiate S&F transformations involving entries in their views and also involving message loss. Each such transformation decreases the source node's outdegree from s to $s-2$ without affecting the outdegree of any other node. After at most n such transformations we get to some $G' \in \mathcal{V}_0$. □

Proof of Lemma 7.1. By Lemmas A.3 and A.4, and since by the definition of $\mathcal{G}$ all states in $\mathcal{V}_1$ are reachable from some state in $\mathcal{V}_0$, the lemma follows. □

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

REFERENCES

[1] A. Allavena, *On the Correctness of Gossip-Based Membership Protocols*, Ph.D. thesis, Cornell University, Ithaca, NY, 2006.

[2] A. Allavena, A. Demers, and J. E. Hopcroft, *Correctness of a gossip based membership protocol*, in Proceedings of the Twenty-Fourth Annual ACM Symposium on the Principles of Distributed Computing, 2005, pp. 292–301.

[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 Automata, Languages and Programming, Lecture Notes in Comput. Sci. 5125, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 121–132.

[4] O. Bakr and I. Keidar, *Evaluating the running time of a communication round over the Internet*, in Proceedings of the Twenty-First Annual ACM Symposium on the Principles of Distributed Computing, 2002, pp. 243–252.

[5] Z. Bar-Yossef, R. Friedman, and G. Kliot, *RaWMS-Random walk based lightweight membership service for wireless ad hoc networks*, ACM Trans. Comput. Syst., 26 (2008), pp. 1–66.

[6] F. Bonnet, *Performance Analysis of Cyclon, an Inexpensive Membership Management for Unstructured P2P Overlays*, Master's thesis, ENS Cachan Bretagne, University of Rennes, IRISA, Rennes, France, 2006.

[7] E. Bortnikov, M. Gurevich, I. Keidar, G. Kliot, and A. Shraer, *Brahms: Byzantine resilient random membership sampling*, Comput. Netw., 53 (2009), pp. 2340–2359.

[8] P. Brémaud, *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*, Springer-Verlag, New York, 2008.

[9] Y. Busnel, M. Bertier, and A.-M. Kermarrec, *Bridging the Gap between Population and Gossip-based Protocols*, Research report RR-6720, INRIA, Rennes, France, 2008.

[10] C. Cooper, M. E. Dyer, and C. S. Greenhill, *Sampling regular graphs and a peer-to-peer network*, Combin. Probab. Comput., 16 (2007), pp. 557–593.

[11] C. Cooper, M. E. Dyer, and A. J. Handley, *The flip Markov chain and a randomising P2P protocol*, in Proceedings of the Twenty-Eighth Annual ACM Symposium on the Principles of Distributed Computing, 2009, pp. 141–150.

[12] D. Dolev, C. Dwork, and L. J. Stockmeyer, *On the minimal synchronism needed for distributed consensus*, J. ACM, 34 (1987), pp. 77–97.

[13] P. Th. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, and A.-M. Kermarrec, *Lightweight probabilistic broadcast*, ACM Trans. Comput. Syst., 21 (2003), pp. 341–374.

[14] T. Feder, A. Guetz, M. Mihail, and A. Saberi, *A local switch Markov chain on given degree graphs with application in connectivity of peer-to-peer networks*, in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, 2006, pp. 69–76.

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

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

[17] 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, Springer-Verlag, Berlin, Heidelberg, 2001, pp. 44–55.

[18] D. Gavidia, S. Voulgaris, and M. van Steen, *Epidemic-Style Monitoring in Large-Scale Sensor Networks*, Technical report IR-CS-012, Vrije Universiteit, Amsterdam, The Netherlands, 2005.

[19] C. Gkantsidis, M. Mihail, and A. Saberi, *Random walks in peer-to-peer networks*, in INFOCOM 2004, Proceedings of the Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies, 2004, pp. 120–130.

[20] J. Gray, *Notes on data base operating systems*, in Operating Systems, an Advanced Course, Springer-Verlag, London, 1978, pp. 393–481.

[21] M. Gurevich and I. Keidar, *Correctness of gossip-based membership under message loss*, in Proceedings of the Twenty-Eighth Annual ACM Symposium on the Principles of Distributed Computing, 2009, pp. 151–160.

[22] S. Hu and W.-Y. Yan, *Stability robustness of networked control systems with respect to packet loss*, Automatica J. IFAC, 43 (2007), pp. 1243–1248.

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

[24] D. S. Lun, M. Madard, R. Koetter, and M. Effros, *On coding for reliable communication over packet networks*, Phys. Commun., 1 (2008), pp. 3–20.

[25] C. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, *Search and replication in unstructured peer-to-peer networks*, in Proceedings of the 16th International Conference on Supercomputing, ACM, New York, 2002, pp. 84–95.

[26] P. Mahlmann and C. Schindelhauer, *Peer-to-peer networks based on random transformations of connected regular undirected graphs*, in Proceedings of the Seventeenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, 2005, pp. 155–164.

[27] P. Mahlmann and C. Schindelhauer, *Distributed random digraph transformations for peer-to-peer networks*, in Proceedings of the Eighteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, 2006, pp. 308–317.

[28] L. Massoulie, E. Le Merrer, A.-M. Kermarrec, and A. J. Ganesh, *Peer counting and sampling in overlay networks: Random walk methods*, in Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, 2006, pp. 123–132.

[29] R. Melamed and I. Keidar, *Araneola: A scalable reliable multicast system for dynamic environments*, J. Parallel Distrib. Comput., 68 (2008), pp. 1539–1560.

[30] B. Morris and Y. Peres, *Evolving sets, mixing and heat kernel bounds*, Probab. Theory Related Fields, 133 (2005), pp. 245–266.

[31] J. R. Norris, *Markov Chains*, Cambridge University Press, Cambridge, UK, 1998.

[32] 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 (1999), pp. 289–299.

[33] N. Tölgyesi and M. Jelasity, *Adaptive peer sampling with newscast*, in Proceedings of the 15th International Euro-Par Conference on Parallel Processing, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 523–534.

[34] S. Voulgaris, D. Gavidia, and M. van Steen, *CYCLON: Inexpensive membership management for unstructured P2P overlays*, J. Netw. Syst. Manag., 13 (2005), pp. 197–217.