Probabilistic reliable dissemination in large-scale systems

Anne-Marie Kermarrec, Laurent Massoulié and Ayalvadi J. Ganesh*

Abstract

The growth of the Internet raises new challenges for the design of distributed systems and applications. In the context of group communication protocols, gossip-based schemes have attracted interest as they are scalable, easy to deploy and resilient to network and process failures. However, traditional gossip-based protocols have two major drawbacks: *(i)* they rely on each peer having knowledge of the global membership and *(ii)* being oblivious to the network topology, they can impose a high load on network links when applied to wide-area settings.

In this paper, we provide a theoretical analysis of gossip-based protocols which relates their reliability to key system parameters (system size, failure rates and number of gossip targets). The results provide guidelines for the design of practical protocols. In particular, they show how reliability can be maintained while alleviating drawback *(i)* by providing each peer with only a small subset of the total membership information, and drawback *(ii)*

*Anne-Marie Kermarrec, Laurent Massoulié and Ayalvadi J. Ganesh are with Microsoft Research, 7 J J Thomson Avenue,Cambridge CB3 0FB, UK. E-mail:{*annemk,lmassoul,ajg*}*@microsoft.com

by organising members into a hierarchical structure that reflects their proximity according to some network-related metric.

We validate the analytical results by simulations, and verify that the hierarchical gossip protocol considerably reduces the load on the network compared to the original, non-hierarchical protocol.

Keywords scalability, reliability, gossip-based probabilistic multicast, membership, group communication, random graphs.

1 Introduction and background

Large-scale reliable group communication Reliable group communication protocols are essential for distributed systems and applications such as publish/subscribe systems [8], distributed databases [6], consistency management [13] and distributed failure detection [26]. The growth of the Internet has influenced the scale and the reliability requirements of distributed systems. Traditional solutions applicable in small-scale settings often do not scale well to very large system sizes.

Network layer multicast protocols like SRM [9] and RMTP [19] work on top of IP multicast [5] and ensure reliability by using positive or negative acknowledgments to repair packet losses. However, IP multicast is not currently deployed in the Internet. Consequently, application-level multicast [17] has recently received increasing attention. Centralized or partially-centralized approaches, proven efficient in local-area networks [3, 18], do not scale well to large groups. For instance, log-based reliable multicast (LBRM) [16] uses *loggers* to provide stable storage and to handle retransmission of missing messages; however, the amount of information to be

stored grows with the number of nodes, and loggers could be overloaded. Other protocols, like Scribe [4] and CAN-multicast [23], are efficient and scalable but require the existence of a large-scale peer-to-peer routing infrastructure. In contrast, epidemic or gossip-based protocols scale well to large groups, are easy to deploy and degrade gracefully as the rate of node failure or message loss increases.

Gossip-based probabilistic multicast protocols These protocols rely on a peer-to-peer interaction model for multicasting a message and are scalable since the load is distributed among all participating nodes. They use redundant messages to achieve reliability and fault tolerance. This class of protocols has been used for consistency management in replicated databases [6, 13], failure detection [26], garbage collection [14] etc. A recent protocol called *pbcast* [2] uses them for reliable multicast. In *pbcast*, notifications are first broadcast using either IP multicast or a randomly generated multicast tree if IP multicast is not available. In addition, each node periodically chooses a random subset of processes and sends them a digest of the most recent messages. Upon receipt of these messages, receivers check for missing messages and, if needed, solicit retransmission.

A related protocol, combining both push and pull phases, is proposed in [22]. In the push phase, each node receiving a message passes it on as in gossip, but increments a counter attached to the message. When the counter reaches a threshold, receiving nodes don't gossip anymore. In the second phase, nodes which haven't yet received the message send requests to randomly chosen nodes to pull the message. This is one of the few papers to include a theoretical analysis of the number of gossip messages needed to ensure high probability of reaching everyone. The pull phase is difficult to implement, which motivates us to consider a pure push algorithm in

this paper. Our analysis techniques are different and may be of independent interest.

A hybrid version of *pbcast* and LBRM, called *Reliable Probabilistic Multicast* (*rpbcast)* [24] consists of three phases. The first phase uses an unreliable IP multicast. During the second phase, a pbcast gossiping step is initiated and, if it fails, a third deterministic phase using loggers is invoked.

A refinement of probabilistic gossip that takes network topology into account is *Directional Gossip* [20]. This is a wide-area protocol in which nodes favour the choice of low connectivity neighbours as gossip targets in an attempt to improve reliability.

Though the above gossip-based approaches have proven scalable, they rely on a non-scalable membership protocol: they assume that the subset of nodes is chosen uniformly among all participating nodes, requiring that each node should know every other node. This assumption limits their applicability in large-scale settings. A protocol which partially addresses this issue is presented in [21], where a connection graph called a Harary graph is constructed. Optimality properties of Harary graphs ensure a good trade-off between the number of messages propagated and the reliability guarantees. However, building such a graph requires global knowledge of membership, and maintaining such a graph structure in the presence of arrivals/departures of nodes might prove difficult.

Contributions In this paper, we consider protocols where each node maintains only a partial view of the membership, which is typically much smaller than the system size. Gossip targets are chosen from this partial view. We describe how probabilistic gossip-based algorithms can be modelled using random graphs, and provide a rigorous analysis relating the probability of success to the fanout, defined as the number of gossip targets per node. We do this first in a *flat*

setting, where the partial views provided to each node are chosen uniformly among all group members. We then consider models where nodes are grouped into clusters and partial views are largely restricted to nodes within the same cluster. We give an analysis of the reliability as a function of the fanout both within and between clusters. This can be extended to hierarchical models with an arbitrary number of levels. The network load can be reduced by clustering on the basis of proximity in the network.

Simulations confirm the analysis in showing that the protocol exhibits stable behavior even under high failure rates. An implication of this resilience to node failures is that the protocol can provide good support for mobile nodes which may disconnect for non-negligible periods. In order to evaluate the impact of clustering, we simulate our system on a realistic network topology based on the Georgia-Tech transit-stub model [27]. Results show that the hierarchical protocol substantially reduces network load compared to flat gossip. The theoretical analysis in this paper is generic and can be applied to a variety of gossip-based protocols.

We describe a generic gossip-based protocol and membership mechanisms in Section 2. The theoretical analysis of the flat and hierarchical protocols are presented in Section 3. Section 4 displays our simulation results. We conclude in Section 5.

2 Gossip and membership protocols

2.1 Gossip protocol

We consider a system composed of n nodes. We assume that there is a membership protocol which provides each node with a randomized partial knowledge of the system; this consists of a list of node identifiers stored in a *local subscription list* whose size we denote by l. We will

consider specific examples of membership protocols later. A *notification* (or gossip) message contains an event to disseminate to the whole group. When a node generates a notification event, a gossiping protocol round is initiated. The pseudo-code for the gossiping algorithm is presented in Algorithm 1. A node that initiates a notification or receives it for the first time picks k nodes at random from its local list and sends them the notification. The number of gossip targets, k, is called the fanout. The numbers k and l could be random variables.

1 Probabilistic gossiping algorithm at each node

```
Receive gossip(sender, notification);
if notification.getId() ∉ historyIdList then
   {check if the node has already received the notification}
   deliver(message);
   historyList.add(notification); {Add the new notification to the history and remove the oldest one}
   for (i=0; i< k; i++) do
      choose target at random from localSubscriptionList; {in practice we ensure that we have k distinct
      random choices. If k = l, all nodes are chosen }
      send(target, myself, notification, "gossip");
   end for
end if
```

The links between nodes defined by their gossip targets specify an overlay network on top of the existing network topology. We call this overlay network, a connection graph in the rest of the paper. In the next section, we derive an expression for the fanout required to achieve a specified probability that a notification reaches every group member. The membership protocol can be tuned to provide members with a partial view which is some small multiple of this desired fanout. This permits nodes to randomize their choice of gossip targets between successive gossip rounds, and reduces the likelihood of a node remaining isolated for long periods.

For the sake of simplicity, we assume that a gossiping round is initiated for each notification. This can easily be modified to be initiated periodically and to send several notifications per gossip message.

2.2 Reliability requirements

The goal of our protocol is to ensure that a notification sent by a member of the group reaches all non-failed members despite transient or permanent failure of other nodes and/or links in the network. Transient failures refer to the temporary inability of a node to receive a message (e.g., due to buffer overflow) or a temporary failure of the network to deliver a message (e.g., due to packet drops). Permanent failures refer to node crashes.

Gossip-based protocols provide probabilistic guarantees of delivery. The parameters of the model can be tuned to achieve success probabilities arbitrarily close to 1, so that our approach is comparable to deterministic methods. In this paper, we focus on the relationship between the fanout and the reliability of the basic gossip protocol described in the previous section. Additional mechanisms to increase reliability can be easily layered on top of the basic probabilistic protocol. This is out of the scope of this paper.

Finally, note that a node may become isolated either because its identifier is present in no local views or because all nodes holding its identifier have either failed or unsubscribed. Such a node has a substantial probability of remaining isolated for a long period. We describe how to deal with this issue in the context of specific membership protocols below.

2.3 Membership protocols

A variety of protocols can be used to provide each node with a partial view of the group membership. The focus here is not on the details of their implementation, but on the theoretical analysis and simulation results in the next two sections, which show that the memory requirements of these protocols scale well in the system size. The results are applicable to variants of

the basic protocols presented below.

2.3.1 Flat membership protocol

Server-based protocol Consider a set of *s* servers[1], to one of which group members have to subscribe when they join a group. Each server manages a *subscription list* containing all subscriptions it knows about. In this model, each server manages a part of the membership service. The subscription process is distributed among the servers whereas the membership information is replicated on all servers. Upon receipt of a subscription, a server adds the subscriber in its own subscription list. The server integrates the new member in the connection graph of the group. This requires two steps: *(i)* providing the new member with a partial knowledge of the system and *(ii)* disseminating the new member's identity to other nodes. To this end, the server randomly chooses a subset of *l* nodes from its subscription list and sends the subset to the new member. This subset will constitute its local subscription list and provides it with a uniform randomized partial view of the system. In addition, the server randomly chooses *l* other nodes and sends them the identifier of the new member. This enables the new member to be integrated in *l* other local subscription lists and consequently in future connection graphs.

As the fanout is related to the number of nodes in the system and the reliability guarantees, servers are in charge of modifying the fanout in response to changes in the number of nodes; such modifications are likely to be infrequent since, as we shall see, the fanout needs to be increased by 1 when the number of nodes increases by a factor of *e*.

The main drawback of this protocol is that, as the number of nodes in the system increases, the load on each server increases linearly. In addition, synchronisation between servers is re-

[1]Many distributed applications still rely on a set of servers for a variety of purposes.

quired periodically to ensure that they have a (approximately) consistent view of the group membership. Replicating membership information has the advantage that failure of individual servers can be tolerated.

Isolation could happen in this protocol when a node's identifier is present in no local views but that of its server, for example because all nodes holding its identifier have either failed or unsubscribed. To overcome this, nodes periodically send heartbeat messages according to the same gossip protocol. Missed heartbeat trigger resubscriptions.

Decentralized flat membership protocols The protocols described above rely on servers to manage the membership, though not the dissemination of notifications. The problem of designing a scalable, peer-to-peer membership service is addressed by Lpbcast [8] and Scamp [10, 11]. Scamp employs a self-organizing subscription mechanism which automatically provides each node with a partial view of the membership of size $(c+1)\log(n)$ on average, where n is the number of members and c a design parameter.

2.3.2 Hierarchical membership protocol

Probabilistic gossip-based protocols are scalable from the nodes' point of view. However, their attractive reliability properties derive from a high degree of redundancy. This generates a large number of messages which may be expensive in a wide-area setting.

This drawback can be partially overcome if most messages are sent locally. In the flat membership approach, the local subscription list is composed of nodes located all over the network. We now introduce a hierarchical model where nodes are clustered according to a

geographical or network proximity criterion[2], and this is taken into account in providing nodes with a local subscription list composed exclusively of nodes belonging to the same cluster. In addition, a few nodes within each cluster are provided with a remote subscription list consisting of nodes in other clusters. We will see that only a small number of links between clusters are necessary to keep the system connected.

We henceforth distinguish between *inter-cluster fanout* and *intra-cluster fanout*:

- The intra-cluster fanout k denotes the number of links each node has with other nodes in the same cluster.
- The inter-cluster fanout f denotes the number of remote links each *cluster* must maintain with nodes outside the cluster. This is the minimum degree of knowledge each server must have of nodes outside its cluster.

The intra-cluster membership information is contained in the subscription list as previously. In addition a *Remote list* contains the identity of f remote nodes[3]. f random nodes are designated in each cluster to maintain these remote links; one node may be responsible for one inter-cluster connection only. Figure 1 depicts an example of a 24-node system with the two different approaches.

The membership protocol has been described for a two-level hierarchy for ease of exposition, but can easily be extended to a hierarchy with more levels. In the model described, remote links are equally likely to be directed at any other cluster, though some clusters may be closer

[2]The geographical criterion used for clustering could be the number of hops or the round-trip delay, for example. It is a challenging problem to estimate these quantities reliably and to locate the nearest server, and this is the subject of much ongoing research.

[3]It is not relevant to the analysis whether these f links go to f distinct clusters or whether a cluster is chosen uniformly at random for each link, so that more than one link could go to the same cluster.

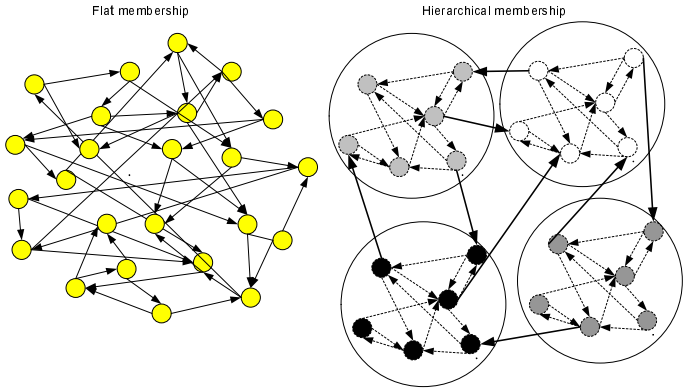

Figure 1: Flat and hierarchical membership: graph of connection in a 24 node system. In the flat model, the fanout is set to 2. In the hierarchical model, both the intra- and inter-cluster fanouts – the latter being represented by the thick arrows–, are set to 2.

than others in the chosen metric. In view of the fact, established in the next section, that very few inter-cluster links are required, we do not expect this to have a significant impact on network load in all but very large systems. If required, differences in proximity between clusters can be exploited by adding more levels to the hierarchy.

The membership protocol can be implemented using a server per cluster which is in charge of attributing inter-cluster links as well as maintaining the partial views within the cluster. Synchronisation is required between servers to exchange random node-Ids which will be used to link clusters.

A hierarchical gossiping approach is used in Astrolabe [25] and [15]. In these protocols, the hierarchy of nodes is mapped on the network topology by an administrator and nodes are more likely to gossip within their sub-hierarchy than remotely. A fully decentralized hierarchical approach to membership has been proposed in Hi-Scamp [12] which clusters nodes according to a network proximity metric and implements a Scamp[11] protocol within each cluster and between clusters at each level of the hierarchy.

3 Analysis of gossiping performance

In this section we establish theoretical results on the performance of the gossiping algorithm, in terms of the following key parameters: fanout, failure rate (both link and node), and system size. To this end we first derive analytical results on the connectivity of random graphs that hold as the number of nodes grows large. We then apply these results to analyse the performance of the flat and hierarchical protocols described above.

3.1 Connectivity of random graphs

Let $G(n,p)$ denotes the random graph with n nodes where, for every ordered pair of nodes $\{x,y\}$, the arc (x,y) (directed from x to y) is present with probability p, independent of every other arc. The presence of arc $\{x,y\}$ is interpreted as saying that y is one of x's gossip targets[4]. The success of the gossip protocol corresponds to the existence of a directed path from a specified source node s to every other node in the random graph.

The connectivity of undirected graphs was studied in a classical paper of Erdös and Renyi [7]. They consider the graph on n nodes where the edge between each (unordered) pair of nodes is present with probability p_n independent of other edges. They show that if $p_n = (\log n + c + o(1))/n$, then the probability that the graph is connected goes to $\exp(-\exp(-c))$. The problem we study is an analogue of this for directed graphs.

Consider the random graph $G(n,p_n)$ with a specified source node s. We denote by $\pi(p_n,n)$ the probability that there is a directed path from s to every other node of $G(n,p_n)$. Likewise,

[4]This model would correspond to our gossiping protocol if the number of gossip targets chosen by each node followed a binomial distribution $Bin(n,p)$. Our results on the success of gossip continue to hold if the number of gossip targets is a constant with the same mean np. We prefer to work with the $G(n,p)$ model because it allows easier analysis of the impact of failures.

given a subgraph of $G(n,p_n)$ with j nodes including s, we denote by $\pi(p_n,j)$ the probability that each of these nodes is reachable from s along the arcs of the subgraph.

Theorem 1 *Consider the sequence of random graphs $G(n,p_n)$ with $p_n = [\log n + c + o(1)]/n$, where c is a constant. We have $\lim_{n\to\infty}\pi(p_n,n) = e^{-e^{-c}}$.*

In fact, this result is applicable more generally, to random directed graph models where the fanout k_n is a random variable with a general distribution. As in the examples above, the probability of having a directed path from the source to all vertices depends only on the mean fanout $\mathbb{E}[k_n]$ and not on the exact distribution of k_n, provided this distribution satisfies certain restrictions. These restrictions are not easy to state and we do not pursue this further here, but refer the reader to Ball and Barbour [1] for details. The proof techniques presented here are different from those in [1] and may be of independent interest.

Proof The proof relies on the identity

$$\pi(p_n,n) = 1 - \sum_{r=1}^{n-1} \binom{n-1}{r-1} (1-p_n)^{r(n-r)} \pi(p_n,r) = 1 - \sum_{r=1}^{n-1} (1-\frac{r}{n}) \binom{n}{r} (1-p_n)^{r(n-r)} \pi(p_n,n-r). \tag{1}$$

To see this, we note that reaching all n vertices is the complement of reaching exactly r vertices, for some r between 1 and $n-1$. For each fixed r, the first term in the sum on the right is the number of ways of choosing $r-1$ vertices other than the source, the second is the probability that there is no arc from any of these r vertices to any of the remaining $n-r$, and the third term is the probability that all r vertices are reached from the source conditional on there being no edges from any of them to outside. The second equality is obtained by simple manipulation. We first briefly sketch the intuition behind the rest of the proof. We can show

that for p_n having the form in the statement of the theorem,

$$\lim_{n\to\infty} \binom{n}{r} (1-p_n)^{r(n-r)} = \frac{e^{-cr}}{r!}$$

for each fixed r. Assuming that $\pi = \lim_{n\to\infty} \pi(p_n, n)$ exists and is the same as $\lim_{n\to\infty} \pi(p_n, n-r)$ for fixed r, and that sums and limits are interchangeable in (1), we expect to have $\pi = 1 - \sum_{r=1}^{\infty} \frac{e^{-cr}}{r!}\pi$. On simplifying, this gives $\pi = \exp(-\exp(-c))$.

We now proceed with the formal proof. Call a vertex isolated if it has no incoming arcs. Clearly, there is a directed path from the source to every other vertex only if there are no isolated vertices (other than, possibly, the source itself). Now, for each vertex $x \neq s$, we have

$$\mathbb{P}(x \text{ is isolated}) = (1-p_n)^{n-1} = \frac{e^{-c}}{n}[1+o(1)].$$

Moreover, under the random graph model $G(n, p_n)$, the isolation of distinct vertices are *independent* events. Thus,

$$\mathbb{P}(\text{no vertex other than } s \text{ is isolated}) = \left(1 - \frac{e^{-c}}{n}[1+o(1)]\right)^{n-1} = e^{-e^{-c}}[1+o(1)].$$

Recalling that $\pi(p_n, n)$ denotes the probability that every vertex is reachable from the source via a directed path, it is immediate from the above that

$$\limsup_{n\to\infty} \pi(p_n, n) \leq e^{-e^{-c}}. \tag{2}$$

In fact, this simple calculation essentially yields the correct estimate of the probability of

there being a directed path to every vertex. In other words, isolated vertices constitute the main contribution to the probability of a vertex not being reachable from the source.

We now establish the reverse inequality to (2) and, thereby, the claim of Theorem 1. In order to establish a lower bound on $\pi(p_n, n)$, the LHS of (1), we will require upper bounds on $\binom{n}{r} p_n^r (1-p_n)^{n-r}$ and $\pi(p_n, n-r)$, and these should be tight enough to yield a satisfactory upper bound on the sum over r of the series on the RHS. The following lemmas establish the necessary upper bounds.

Lemma 2 *Let $np_n = \log n + c + o(1)$. Then, for all n sufficiently large, and all $r \in \{1, \ldots, n/2\}$, we have*

$$f_n(r) := \binom{n}{r} (1-p_n)^{r(n-r)} \leq \frac{e^{-(c-1)r}}{\lfloor r/2 \rfloor !}.$$

Moreover, $\lim_{n\to\infty} f_n(r) = e^{-cr}/r!$ for each fixed r.

Proof We have $n\log(1-p_n) = -\log n - c + o(1)$, and so,

$$\begin{aligned} \log f_n(r) &= -\log r! - r[c + o(1)] + \sum_{j=1}^{r-1} \log(1 - \frac{j}{n}) + \frac{r^2(\log n + c)}{n} \\ &\leq -\log r! - r[c + o(1)] + \frac{r^2(\log n + c)}{n}. \end{aligned}$$

It is clear from the second equality above that, for fixed r, $\lim_{n\to\infty} \log f_n(r) = -\log r! - cr$.

This verifies the second claim of the lemma.

A standard comparison of sums and integrals shows that $\log r! \geq \int_1^r \log x dx$, which in turn implies that $\log r! \geq r\log r - r + 1$ for all $r \geq 1$. Hence, for all $r \in \{1, \ldots, \frac{n}{2}\}$ and n sufficiently

large,

$$\log f_n(r) \leq -(c-1)r - r\log r\left[1 - \frac{\log n/n}{\log r/r}\right].$$

It can be verified by differentiation that $\log x/x$ is a decreasing function of x for $x > e$. Hence, for all $r \in \{3,\ldots,n/2\}$,

$$\begin{aligned}\log f_n(r) &\leq -(c-1)r - \frac{r}{2}\log\frac{r}{2} - \frac{r}{2}\log 2\left[1 - \frac{\log r}{\log(n/2)}\right] \\ &\leq -(c-1)r - \log\left(\lfloor\frac{r}{2}\rfloor!\right).\end{aligned}$$

We have used the fact that $\log n! \leq n\log n$ to obtain the last inequality. This establishes the first claim of the lemma for $3 \leq r \leq n/2$. It is straightforward to verify the claim for $r = 1,2$. ■

Lemma 3 *Let f_n be defined as in Lemma 2. Given $\varepsilon > 0$, we can find R such that, for all n sufficiently large, $\sum_{r=R+1}^{n}(1-\frac{r}{n})f_n(r) < \varepsilon$.*

Proof We have from Lemma 2 that, for all $r \in \{1,\ldots,n/2\}$ and n sufficiently large,

$$f_n(r) \leq g(r) := \frac{e^{-(c-1)r}}{\lfloor r/2\rfloor!}.$$

But the positive sequence $g(r)$ is summable, since

$$\sum_{r=0}^{\infty} g(r) \leq \sum_{r=0}^{\infty}\frac{1}{r!}[e^{-(c-1)2r} + e^{-(c-1)(2r+1)}] \leq (1+e^{-(c-1)})e^{e^{-2(c-1)}}.$$

Hence, given $\varepsilon > 0$, we can choose R large enough that, for all n sufficiently large,

$$\sum_{r=R+1}^{n/2} f_n(r) \leq \sum_{r=R+1}^{\infty} g(r) < \frac{\varepsilon}{3}.$$

Since $f_n(r) = f_n(n-r)$, it follows that $\sum_{r=R+1}^{n-R-1}(1-\frac{r}{n})f_n(r) < \frac{2\varepsilon}{3}$. Let $g_{\max} = \max\{g(r) : r \geq 0\}$ and note that it is finite. Thus, $\sum_{r=n-R}^{n}(1-\frac{r}{n})f_n(r) \leq \frac{Rg_{\max}}{n}$. Since R and $g_{\max}$ don't depend on n, the last term is smaller than $\varepsilon/3$ for all n sufficiently large. This completes the proof of the lemma. ■

Lemma 4 *Let $R \in \mathbb{N}$ be given. Then, for all n sufficiently large, we have for $r \in \{1,\ldots,R\}$, that*

$$\pi(p_n, n-r) \leq \pi(p_n, n)\big(1 + \frac{2R}{n}e^{-c}\big).$$

Proof Fix $r \in \{1,\ldots,R\}$. Let A be a set of vertices of cardinality $n-r$ containing the source vertex, s. Let x be a vertex not in A and let $B = A \cup \{x\}$. Denote by E_A the event that all vertices in A are reachable from s via directed paths that don't leave A; define E_B analogously. Then, $\mathbb{P}(E_A) = \pi(p_n, n-r)$, $\mathbb{P}(E_B) = \pi(p_n, n-r+1)$ and we have

$$\mathbb{P}(E_B) \geq \mathbb{P}(E_A \cap E_B) = \mathbb{P}(E_A)\,\mathbb{P}(E_B|E_A) = \mathbb{P}(E_A)[1 - \mathbb{P}(E_B^c|E_A)],$$

where E_B^c denotes the complement of E_B. But $\mathbb{P}(E_B^c|E_A) = (1-p_n)^{n-r}$ since, conditional on E_A, the event E_B^c is equivalent to there being no arcs from any of the $n-r$ vertices in A to vertex x. Thus, $\pi(p_n, n-r) \leq \pi(p_n, n-r+1)/[1-(1-p_n)^{n-r}]$. Iterating this inequality yields

$$\begin{aligned}
\pi(p_n, n-r) &\leq \frac{\pi(p_n, n)}{\prod_{j=1}^{r}[1-(1-p_n)^{n-j}]} \leq \pi(p_n, n)\left[1-(1-p_n)^{n-R}\right]^{-R} \\
&\leq \pi(p_n, n)\big(1 - \frac{e^{-c}}{n}[1+o(1)]\big)^{-R} \leq \pi(p_n, n)\big(1 + \frac{R}{n}e^{-c}[1+o(1)]\big).
\end{aligned}$$

The claim of the lemma follows for large enough n. ■

We now have from (1) and Lemmas 2, 3 and 4 that

$$\begin{aligned}\pi(p_n,n) &\geq 1-\sum_{r=1}^{R}(1-\frac{r}{n})\,f_n(r)\pi(p_n,n-r)-\sum_{r=R+1}^{n-1}(1-\frac{r}{n})\,f_n(r)\\ &\geq 1-\left(1+\frac{2R}{n}e^{-c}\right)\sum_{r=1}^{R}(1-\frac{r}{n})\,f_n(r)\pi(p_n,n)-\varepsilon.\end{aligned}$$

In other words,

$$\pi(p_n,n)\geq(1-\varepsilon)\left[1+\left(1+\frac{2R}{n}e^{-c}\right)\sum_{r=1}^{R}(1-\frac{r}{n})\,f_n(r)\right]^{-1}.$$

Letting $n\to\infty$ with R and ε fixed, we obtain using Lemma 2 that

$$\liminf_{n\to\infty}\pi(p_n,n)\geq(1-\varepsilon)\left[\sum_{r=0}^{R}\frac{e^{-cr}}{r!}\right]^{-1}.$$

Since we can choose ε arbitrarily small and R arbitrarily large, we get

$$\liminf_{n\to\infty}\ \pi(p_n,n)\ \geq\ e^{-e^{-c}}. \tag{3}$$

Combined with the reverse inequality established in (2), this completes the proof of the theorem.

3.2 Flat Gossip

The result of Theorem 1 is directly applicable to a flat membership model, where it gives the probability of success of the gossip protocol. Intuitively the meaning of the theorem is that there is a sharp threshold in the required fanout at $\log(n)$.

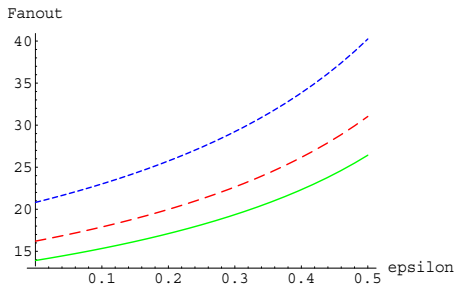

Figure 2: Fanout required versus probability of node failure, for n=1000,10000,100000.

We now consider the impact of link and node failures on the success of the gossiping algorithm. Suppose links fail independently of each other, each with a probability ε. This simply corresponds to replacing p_n by $(1-\varepsilon)p_n$. Therefore, if we take $p_n = [\log n + c + o(1)]/(1-\varepsilon)n$, then the probability of success is asymptotically $\exp(-\exp(-c))$.

Suppose next that nodes other than the source fail independently of the arcs present. The question is then whether the message reaches every node that has not failed. Let us condition on n' being the number of nodes that haven't failed. By the independence assumption, the random graph model for this situation is $G(n', p_n)$. If $p_n = [\log n' + c + o(1)]/n'$, which corresponds to a fanout of

$$k = (n/n')[\log n' + c + o(1)], \tag{4}$$

then gossip succeeds with limiting probability $\exp(-\exp(-c))$. Let ε be the proportion of failed nodes, so that $n' = (1-\varepsilon)n$. Figure 2 shows the mean fanout required to achieve success probability greater than 99.9% as a function of the proportion of failed nodes ε. From top to bottom, the three plots correspond to $n = 100000, 10000, 1000$ respectively.

3.3 Hierarchical gossip

In order for gossip to succeed in this model, it is sufficient that gossip succeeds within each cluster, and also that the message goes from its originating cluster to every other cluster. Let there be N distinct clusters with m nodes in each cluster. Let $n = mN$ denote the total number of nodes. We want to choose the intra-cluster fanout k and the inter-cluster fanout f in order to guarantee bounds on the probability of success.

Let π_{intra} denote the probability that a gossip initiated within a cluster reaches all nodes in that cluster using only intra-cluster arcs. We have from Theorem 1 that, if $k = \log(m) + c_1$, then π_{intra} goes to $\exp(-\exp(-c_1))$ as m goes to infinity. Consider now the graph where each cluster is reduced to a single node, and these nodes are connected by the inter-cluster arcs. If the number of inter-cluster links is binomial $Bin(N, f/N)$, then this is a random graph $\boldsymbol{G}(N, f/N)$[5]. Let π_{inter} denote the probability that there is a directed path from the source cluster to every other cluster in this random graph. Then Theorem 1 says that, if $f = \log(N) + c_2$, then the limiting value of π_{inter} is $\exp(-\exp(-c_2))$ as N grows large.

The overall probability of success $\pi_{success}$ is larger than $\pi_{intra}^N \times \pi_{inter}$. The asymptotic estimates obtained above yield the lower bound $\exp(-Ne^{-c_1} - e^{-c_2})$ on $\pi_{success}$ as m and N go to infinity, if f and k are chosen as above. In order to achieve a target $\pi_{success}$ at least $e^{-\beta}$, we can for instance set $Ne^{-c_1} = e^{-c_2} = \beta/2$. This yields $c_2 = -\log(\beta/2)$ and $c_1 = \log(N) - \log(\beta/2)$, from which

$$f = \log(N) - \log(\beta/2), \text{ and } k = \log(mN) - \log(\beta/2).$$

Observe that the required intra-cluster fanout depends only on mN, the total number of nodes

[5] If there can be multiple links between the same pair of clusters, then the edge probability in the random graph is in fact $1 - (1 - 1/(N-1))^f$, which is smaller, but equivalent to f/N in the asymptotic regime we consider.

in the system. These estimates can be modified to account for node and link failures as for flat gossip.

The lower bound on $\pi_{success}$ can alternatively be expressed as $\pi_{success} \geq \exp(-ne^{-k} - Ne^{-f})$. Figure 3 illustrates the behaviour of this quantity for $n = 10,000$ and $N = 100$, as a function of the fanouts k, f. The light contours, which correspond to high success probabilities, are approximately L-shaped, indicating that there is a sharp threshold for both intra- and inter-cluster fanouts.

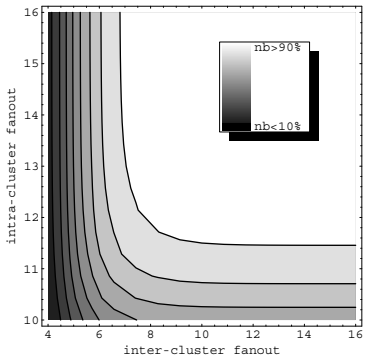

Figure 3: Impact of inter- and intra-cluster fanouts on (lower bound of) success probability in a 100*100-clustered configuration.

4 Simulation results

We evaluate the flat and hierarchical gossip protocols according to the following metrics: *(i)* reliability in failure-free execution; *(ii)* resilience to node failures; *(iii)* the impact on the network in terms of link stress, and *(iv)* the latency of message delivery.

We use a simple packet-level discrete event simulator and run the simulations on network topologies generated randomly according to the Georgia Tech [27] transit-stub model. The topology used in this paper is composed of a 600 node core, with a LAN attached to each core

node. Each LAN has a star topology and is composed of 100 nodes on average. Thus, there are 60,000 LAN nodes. We evaluate two groups, composed of 10,000 and 50,000 nodes selected at random from the 60,000 nodes. Link delays are modeled by simply assigning a propagation delay of 1ms to each LAN link and 40.5ms to each core link. We evaluate the impact of the protocols by measuring the stress on each link of the simulated network, *i.e.*, the number of messages travelling along that link, when one message is broadcast in the group. Simulation results presented below are consistent with the theoretical analysis. The results presented are based on 100 simulations for each configuration.

For hierarchical gossip, we cluster nodes in the following manner. Starting with an arbitrary node, we construct a sorted list by choosing the successor of each node as the one closest to it among nodes not yet in the list. Closeness is measured by delay in the transit-stub topology. Thus nodes belonging to the same LAN are placed at contiguous locations in the list. The sorted listed is then split into equally sized clusters, the number of clusters being a system parameter.

4.1 Probability of atomic broadcast

We report the fraction of simulations in which the broadcast is atomic, by which we mean that every node receives the notification, both with and without failures. We also report the fraction of nodes which receive the notification in non-atomic broadcasts. The corresponding standard deviations are not presented here, but are small and indicate a high degree of confidence in the estimates.

Figures 4 and 5 depict the results in failure-free execution for configurations with 10,000 and 50,000 nodes. In each plot, the dark bar denotes the proportion of simulations where the broadcast was atomic, while the light bar represents the proportion of nodes that received the

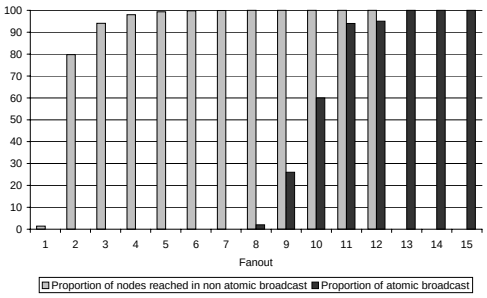

Figure 4: Flat gossip: Results in failure-free execution for 10,000 node group

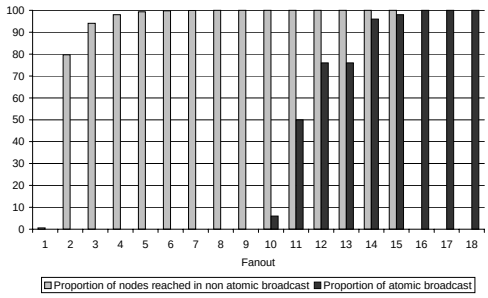

Figure 5: Flat gossip: Results in failure-free execution for 50,000 node group

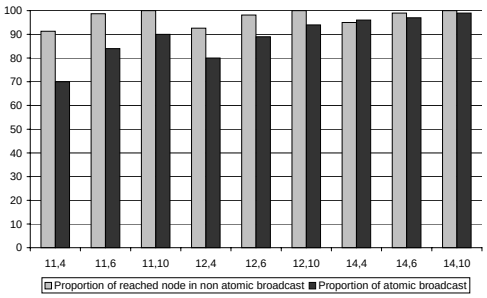

Figure 6: Hierarchical gossip: Results in failure-free execution for 10,000 node in a 10*1000 node group

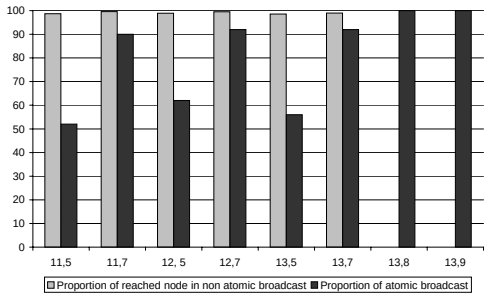

Figure 7: Hierarchical gossip: Results in failure-free execution for 10,000 nodes in a 100*100 node group

message in non-atomic broadcasts. We observe three different regions in the behavior of the system. For low fanouts, atomic broadcast never occurs, and the proportion of reached nodes increases from close to 0 to close to 1. For fanouts in an intermediate range, the proportion of reached nodes remains close to 1, and the proportion of atomic broadcasts increases from 0 to 1. For high enough fanouts, almost all broadcasts are atomic.

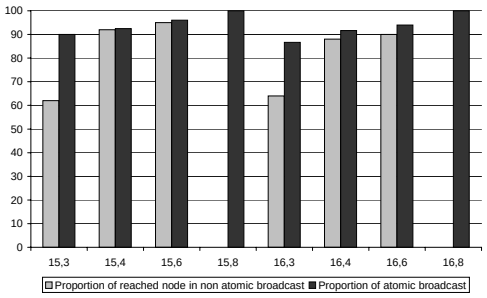

Figure 8: Hierarchical gossip: Results in failure-free execution for 50,000 node in a 10*5000 node group

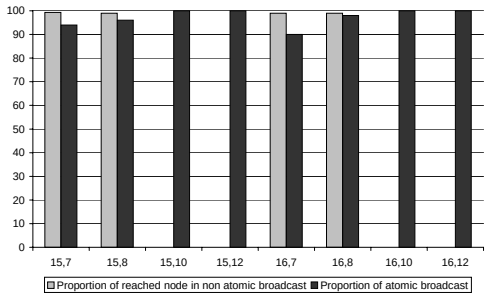

Figure 9: Hierarchical gossip: Results in failure-free execution for 50,000 nodes in a 100*500 node group

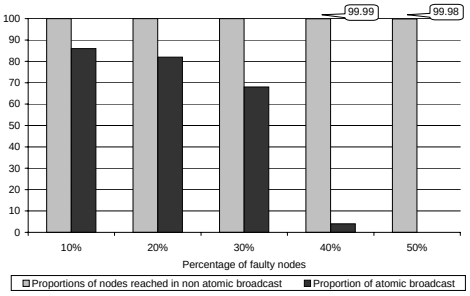

Figure 10: Stability in presence of failures in a 10000 (fanout=13) node group

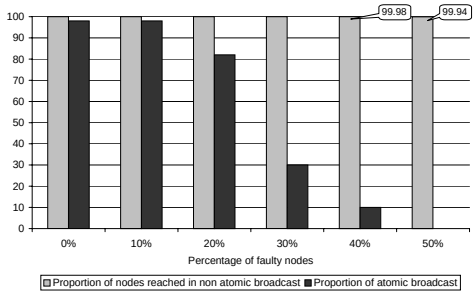

Figure 11: Stability in presence of failures in a 50000 (fanout=15) node group

Figures 6, 7, 8 and 9 display results in failure-free execution for two clustered configurations each for a 10,000-node and a 50,000 node group. Intra- and inter-cluster fanout pairs are indicated on the horizontal axis in this order. The results are consistent with the theoretical analysis.

4.2 Reliability

As mentioned earlier, the performance of gossip-based protocols degrades gracefully in the presence of failures. We illustrate this by evaluating the impact of failures, ranging from 10 to 50% of group members, on both the proportion of atomic broadcast and the proportion of nodes reached by a broadcast message. We consider a fail-stop model where faulty nodes do not gossip messages they receive.

Flat gossiping Figures 10 and 11 display the impact of failures on 10,000-node and 50,000-node configurations with a fixed fanout of 13 and 15 respectively. In failure-free execution with these fanouts, all simulations resulted in an atomic broadcast in the 10,000 node group, and a very large proportion resulted in an atomic broadcast in the 50,000 node group. Figure 10 and 11 show that the proportion of atomic broadcasts decreases substantially when over 20% of

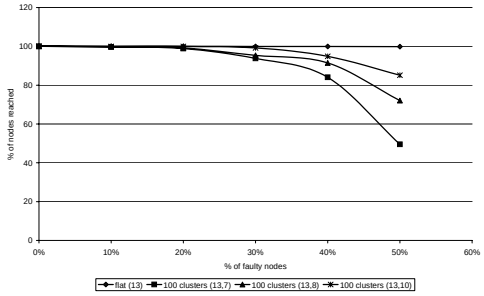

Figure 12: Hierarchical gossiping: Stability in presence of failures in a 10000 node group

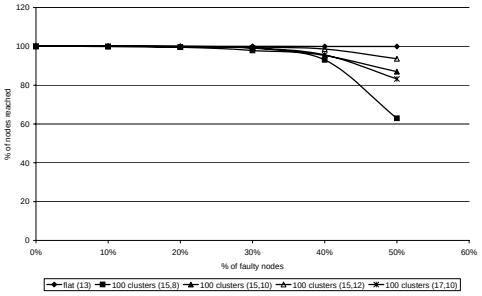

Figure 13: Hierarchical gossiping: Stability in presence of failures in a 50000 node group

group member are faulty. However, the proportion of node reached by a broadcast is still close to 100% even when half of the group is faulty. This demonstrates a high degree of resilience to failures and matches the theoretical results.

Hierarchical gossip The plots (Figures 12 and 13) are very similar to the ones obtained with the flat model, and confirm the resilience of the cluster-based system to failures. The proportion of atomic broadcasts is not shown but decreases sharply as the node failure rate goes up. However, the fraction of nodes reached by a notification remains very high. We also observe that the performance degrades earlier than in the flat version. This is explained by the fact that, if nodes carrying inter-cluster links are faulty, then a whole cluster might be isolated. In contrast, the probability of a large set of nodes being isolated in flat gossip is very low.

4.3 Impact on the network

The main benefit expected from the hierarchical protocol is a decrease in the load on core router links. In order to evaluate this, we compare the load on each physical link, called the link stress, in the flat and the hierarchical protocols. In our experiments, we broadcast one message, and the link stress is defined as the number of copies of this message that traverse the link. We evaluate

	Mean	Median	Max
10,000 nodes (flat)	52	12	12,800
50,000 nodes (flat)	66	14	73,000

Table 1: Link stress characteristics for flat gossip

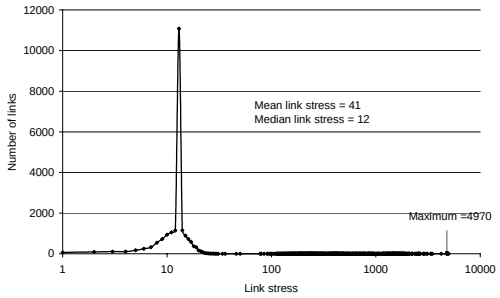

Figure 14: Link stress distribution in a 600 node core topology for a 10*1000 node group (intrafanout=13, interfanout= 4)

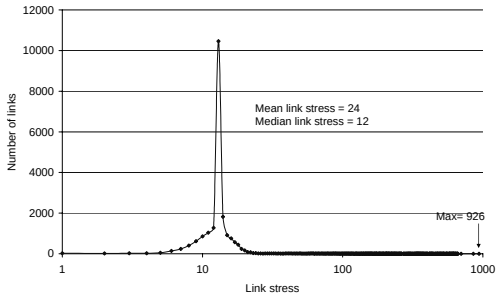

Figure 15: Link stress distribution in a 600 node core topology for a 100*100 node group (intrafanout = 13, interfanout= 7)

the link stress in a non-faulty environment.

Table 1 presents the summary statistics on link stress for flat gossip, for 10,000 and 50,000 nodes with respective fanouts of 12 and 14. In both cases the median coincides with the fanout, a feature explained by the fact that all links within a LAN see exactly this load. As expected, a few links in the core router experience a very high load, due to the network-oblivious choice of gossip targets.

Figures 14, 15, 16, and 17 present the link stress distribution respectively for a 10,000 and 50,000 node group with two different degrees of clustering in each case. The median link stress is still very close to the intra-cluster fanout since most nodes have this fanout. However, we observe a dramatic decrease in both the mean and the maximum link stress, and this is more pronounced as the clusters get smaller. For example in a 10,000 group node, the maximum link stress decreases from 12,800 in the flat version to 4,970 in a 10*1000 clustering to 926 in a 100*100 clustering. Observe that in the latter case, we have approximately one cluster per

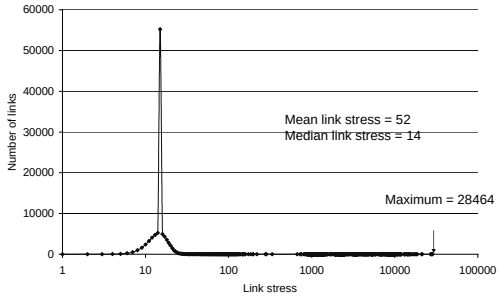

Figure 16: Link stress distribution in a 600 node core topology for a 10*5000 node group (intrafanout=15, interfanout= 4)

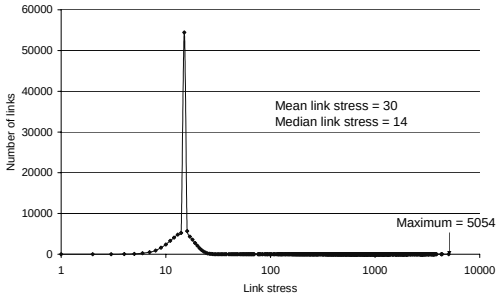

Figure 17: Link stress distribution in a 600 node core topology for a 100*500 node group (intrafanout = 15, interfanout= 7)

LAN, which intuitively minimizes load on the backbone links of the network core. Given that our clustering algorithm is sub-optimal, we can expect a more accurate clustering to have an even greater impact.

4.4 Latency

We also measured the average delay of message delivery for various configurations. We compare the delay for IP multicast (i.e., shortest path delay between source and destination) with that for flat and hierarchical configurations. Table 2 presents the delays for 10,000 node and 50,000 node groups respectively. The fanout is indicated in parentheses. For hierarchical configurations, the intra-cluster fanout is indicated first followed by the inter-cluster fanout. The delay in the hierarchical and flat gossip-based protocols are comparable with a suitable choice of the inter-cluster fanout. In the hierarchical case, the delay decreases as the inter-cluster fanout increases. The delay in the flat version is approximatively three times the IP delay.

Group size	10,000			50,000		
	Fanout	Mean	Max	Fanout	Mean	Max
IP		259	492		259	492
Flat	13	748	1157	15	799	1212
10 clusters	(13,3)	1173	2163	(15,4)	1245	2293
10 clusters	(13,4)	993	1784	(15,7)	1151	2115
100 clusters	(13,7)	904	1602	(15,8)	768	1408
100 clusters	(13,8)	714	1270	(15,12)	662	1211

Table 2: Average and maximum delays in flat and clustered configurations

5 Conclusion

We presented a theoretical analysis of gossip-based protocols relating the fanout and the fraction of link and node failures to the probability of successful information dissemination, for both flat and hierarchical gossiping schemes. Simulations confirmed the theoretical findings and also showed that the hierarchical protocol significantly reduces network link stress at the cost of a small degradation in reliability and latency.

Acknowledgements We would like to thank Miguel Castro and Antony Rowstron for their participation in the development of the simulator.

References

[1] F. Ball and A. Barbour. Poisson approximation for some epidemic models. *Journal of Applied Probability*, 27:479–490, 1990.

[2] K.P. Birman, M. Hayden, O.Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky. Bimodal multicast. *ACM Transactions on Computer Systems*, 17(2):41–88, May 1999.

[3] K.P. Birman and T.A. Joseph. Exploiting virtual synchrony in distributed systems. In *SOSP*, December 1987.

[4] M. Castro, P. Druschel, A-M. Kermarrec, and A. Rowstron. SCRIBE: A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in Communications (JSAC)*, 20(8), October 2002.

[5] S. Deering and D. Cheriton. Multicast routing in datagram internetworks and extended LANs. *ACM Transactions on Computer Systems*, 8(2), May 1990.

[6] A.J. Demers, D.H. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *PODC*, pages 1–12, August 1987.

[7] P. Erdös and A. Renyi. On the evolution of random graphs. *Mat Kutato Int. Közl*, 5(17):17–60, 1960.

[8] P.T. Eugster, R. Guerraoui, S.B. Handurukande, A.-M. Kermarrec, and P. Kouznetsov. Lightweight probabilistic broadcast. In *DSN*, 2001.

[9] S. Floyd, V. Jacobson, C.G. liu, S. McCanne, and L. Zhang. A reliable multicast framework for light-weight sessions and application level framing. *IEEE/ACM Transactions on Networking*, pages 784–803, December 1997.

[10] A.J. Ganesh, A.-M. Kermarrec, and L. Massoulié. SCAMP: Peer-to-peer lightweight membership service for large-scale group communication. In *NGC*, November 2001.

[11] A.J. Ganesh, A.-M. Kermarrec, and L. Massoulié. Peer-to-peer membership management for gossip-based protocols. *IEEE Transactions on Computers*, February 2003. To appear.

[12] A.J. Ganesh, L. Massoulié, and A.-M. Kermarrec. Hi-scamp: self-organizing hierarchical membership protocol. In *SIGOPS European Workshop*, September 2002.

[13] R. Golding and K. Taylor. Group membership in the epidemic style. Technical Report UCSC-CRL-92-13, UC Santa Cruz, Dept. of Computer Science, 1992.

[14] K. Guo, M. Hayden, R. van Renesse, W. Vogels, and K. Birman. GSGC: an efficient gossip-based garbage collection scheme for scalable reliable multicast. Technical Report TR-97-1656, Cornell University, Department of Computer Science, 1997.

[15] I. Gupta, A.-M Kermarrec, and A.J. Ganesh. Adaptive and efficient epidemic-style protocols for reliable and scalable multicast. In *SRDS*, Osaka, Japan, oct 2002.

[16] H. Holbrook, S. Singhal, and D. Cheriton. Log-based receiver-reliable multicast for distributed interactive simulation. In *SIGCOMM*, 1995.

[17] J. Jannotti, D.K. Gifford, K.L. Johnson, F. Kaashoek, and J.W. O'Toole. Overcast: Reliable multicasting with an overlay network. In *OSDI*, San Diego, CA, 2000.

[18] F. Kaashoek, A.S. Tanenbaum, A.S. Hummel, and H.E. Bal. An efficient reliable broadcast protocol. *Operating System Review*, 23, 1989.

[19] J.C. Lin and S. Paul. A reliable multicast transport protocol. In *INFOCOM*, pages 1414–1424, 1996.

[20] M.-J. Lin and K. Marzullo. Directional gossip: Gossip in a wide-area network. Technical Report CS1999-0622, University of California, San Diego, Computer Science and Engineering, June 1999.

[21] M.-J. Lin, K. Marzullo, and S. Masini. Gossip versus deterministic flooding: Low message overhead and high-reliability for broadcasting on small networks. In *14th International Symposium on Distributed Computing*, pages 253–267, 2000.

[22] S. Shenker R. Karp, C. Schindelhauer and B. Vöcking. Randomized rumour spreading. In *IEEE FOCS 2000*, pages 565–574, 2000.

[23] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Application-level multicast using content-addressable networks. In *NGC*, November 2001.

[24] Q. Sun and D.C. Sturman. A gossip-based reliable multicast for large-scale high-throughput applications. In *Proceedings of the International conference on dependable Systems and Networks(DSN2000)*, New York, USA, July 2000.

[25] R. van Renesse and K. Birman. Scalable management and data mining using astrolabe. In *IPTPS*, March 2002.

[26] R. van Renesse, Y. Minsky, and M. Hayden. A gossip-style failure detection service. In *Middleware*, 1998.

[27] E.W. Zegura, K.L. Calvert, and S.Bhattacharjee. How to model an internetwork. In *IEEE Infocom*, April 1996.

Ayalvadi Ganesh graduated from the Indian Institute of Technology, Madras in 1988, and received his M.S. and Ph.D. in Electrical Engineering from Cornell University in 1991 and 1995 respectively. His research interests include queueing theory, congestion control and pricing in the Internet, and distributed systems.

Anne-Marie Kermarrec received the Ph.D degree in Computer science from the University of Rennes, France in 1996. She has worked as a Post-doc Researcher in the Computer Systems Group of Vrije Universiteit in Amsterdam, The Netherlands in 1996-1997 and as an Assistant Professor at the University of Rennes from 1997 to 2000. Since March 2000, she has worked as a Researcher at Microsoft Research, Cambridge, U.K. Her research interests are in distributed systems, fault-tolerance, application-level multicast and peer-to-peer computing.

Laurent Massoulié graduated from the Ecole Polytechnique and received the PhD degree from the Université Paris Sud, Paris France. He is currently a researcher with Microsoft Research, Cambridge, U.K., where he works on modeling and performance analysis of networks. His recent research interests are in quality of service and congestion control for the Internet, and in epidemic-style information dissemination. He is currently associate editor of *Queueing Systems: Theory and Applications*.