SCAMP: Peer-to-peer lightweight membership service for large-scale group communication

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

Microsoft Research Ltd., 7JJ Thomson Avenue Cambridge CB3 0FB, UK
{ajg,annemk,lmassoul}@microsoft.com

Abstract. Gossip-based protocols have received considerable attention for broadcast applications due to their attractive scalability and reliability properties. The reliability of probabilistic gossip schemes studied so far depends on each user having knowledge of the global membership and choosing gossip targets uniformly at random. The requirement of global knowledge is undesirable in large-scale distributed systems.
In this paper, we present a novel peer-to-peer membership service which operates in a completely decentralized manner in that nobody has global knowledge of membership. However, membership information is replicated robustly enough to support gossip with high reliability. Our scheme is completely self-organizing in the sense that the size of local views naturally converges to the 'right' value for gossip to succeed. This 'right' value is a function of system size, but is achieved without any node having to know the system size. We present the design, theoretical analysis and preliminary evaluation of SCAMP. Simulations show that its performance is comparable to that of previous schemes which use global knowledge of membership at each node.

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

1 Introduction

The demand for large-scale event dissemination in distributed systems is growing rapidly but traditional network-level protocols and broadcast algorithms do not scale to more than thousands of participants [9]. Techniques such as SRM (Scalable Reliable Multicast Protocol) [6] or RMTP (Reliable Message Transport Protocol) [9] have added reliability to network-level IP multicast [3, 4] solutions, using acknowledgements and repair mechanisms. However, no feature is available for membership tracking in network-level multicast approaches and their applicability is limited by the lack of wide deployment of IP multicast. As a result, application-level multicast, and in particular, gossip-based broadcast algorithms, have recently emerged as an attractive alternative. Probabilistic versions of these have received much attention and provide good scalability and reliability properties [2, 10, 12]. Their scalability relies on a peer-to-peer interaction model, where each participating node is in charge of a part of the dissemination process: the first time a node receives each notification, it forwards it to a random subset of other

nodes (see the next Section for details). The protocols incorporate redundant messages which make them highly resilient to failures.

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 imposes high requirements on memory and synchronisation, which adversely affects their scalability. This has motivated work on distributing membership management [10, 5] in order to provide each node with a partial random view of the system without any node having global knowledge of the membership.

Our understanding of scalable membership protocol should not be confused with that of [1], [7] where the aim is to provide each member of the group with an accurate and timely global view of the membership. The problem we consider is instead to provide each node with partial membership information which is sufficient to achieve reliable dissemination using a traditional gossip-based protocol. One approach to this issue is presented in [11], 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.

A protocol that does not rely on global knowledge of membership is *Lpbcast* [5]. However, the size of the partial view and the number of gossip targets are fixed *a priori*, which precludes decentralized adaptation to changes in system size.

We seek to provide a fully decentralized membership scheme, which meets the following goals: nodes obtain a partial view that adapts automatically to system size, and the view size is tuned to support gossip-based dissemination. In earlier work [8], we derived the fanout (number of gossip targets) required, as a function of system size, in order to achieve reliability. When the membership management is centralized or distributed among a few servers, the number of participants is easily determined, and the fanout can be adjusted to match reliability requirements. However, in a fully decentralized model, where each node operates with an incomplete view of the system, this is no longer straightforward.

We propose a novel probabilistic scalable membership protocol (SCAMP) aimed at addressing this problem. SCAMP is very simple, fully decentralized and self-configuring. As the number of participating nodes, n, increases, we show both analytically and through simulation that the size of local views automatically adapts to the desired value of $(c+1)\log n$. Here, c is a design parameter which specifies the degree of robustness to failures: it follows from [8, Theorem 1] that any proportion of failed links up to $c/(c+1)$ can be tolerated when the fanout is set to $(c+1)\log n$. Preliminary evaluation results show that gossip based on the partial views provided by SCAMP is as resilient to failures as gossip based on random choice from a global membership known at each node. SCAMP can potentially be incorporated in existing gossip-based schemes to reduce memory and synchronization overhead due to membership management.

The remainder of the paper is organized as follows. We describe SCAMP in Section 2. The theoretical analysis is presented in Section 3 and simulation results in Section 4. We conclude in Section 5.

2 SCAMP: Peer-to-peer lightweight membership service for large-scale group communication

In this section we present the system model and the algorithms of SCAMP. The scalability of the algorithm relies on its peer to peer communication model between nodes for both membership management and gossip dissemination. We have designed SCAMP to achieve partial views of just the right size to be resilient to a given fraction of failures. This presupposes that nodes gossip to all nodes in their partial view. They could choose to gossip to a randomly chosen subset instead at the cost of reducing the fraction of failures tolerated.

In gossip-based protocols, notifications are propagated as follows. When a node generates a notification, it sends it to a random subset of other nodes. When any node receives a notification for the first time, it does the same. The question is how large this random subset should be chosen in order for all nodes to receive the notification with high probability. In earlier work [8], we proved the following result. If there are n nodes, and each node gossips to $\log n + s$ other nodes on average, then the probability that everyone gets the notification converges to $\exp(-e^{-s})$. In other words, there is a sharp threshold at $\log n$: the probability of success (everyone receiving the notification) is close to one if each node gossips to slightly more than $\log n$ nodes and close to zero if each node gossips to slightly fewer than $\log n$ nodes. We also derived expressions for how the success probability depends on the failure rate of nodes and links.

Previous work on gossip-based protocols has relied on each node having knowledge of the global membership list so that gossip targets can be chosen uniformly at random from all members. In [8], we proposed a scheme whereby a set of servers maintains the global membership list and provides individual nodes with a randomized partial view. Thus, nodes don't all need to have global information, but simply gossip to everyone in their local list. In the present work, we eliminate the need for servers and describe a fully decentralized scheme which achieves the same goals: nodes obtain a randomized partial view of the system, and the size of this view automatically scales correctly with system size, even though no node knows the system size. We now describe the details of this scheme.

2.1 Membership management in SCAMP

Subscription New nodes join the group by sending a subscription request to an arbitrary member. They start with a local view consisting of just the member to whom they sent their subscription request. When a node receives a new subscription request, it forwards the new node-id to all members of its own local view. It also creates c additional copies of the new subscription (c is a design parameter that determines the proportion of failures tolerated) and forwards them to randomly chosen nodes in its local view. When a node receives a forwarded subscription (2), it integrates the new subscriber in its view with a probability p which depends on the size of its view. If it doesn't keep the new subscriber, it forwards the subscription to a node randomly chosen from its local view. The system configures itself towards views of size $(c+1)\log(n)$ on average, n being the number of nodes in the system.

Algorithm 1 depicts the pseudo-code for a node receiving a new subscription. Algorithm 2 depicts the pseudo-code for a node receiving a forwarded subscription.

1 Subscription management

```
Upon subscription(s) of a new subscriber

{The subscription of s is forwarded to all the nodes of view}
for (i=0; i< view.Count; i++) do
   {For each node n in View}
   Send(view[i],s,forwardedSusbcription);
end for
{c additional copies of the subscription s are forwarded to random nodes of view}
for (j=0; j< c; j++) do
   randomNode=RandomChoice(view.Count);
   Send(view[randomNode],s,forwardedSusbcription);
end for
```

2 Handling of a forwarded subscription

```
{A node receiving a forwarded subscription adds it with the probability p = 1/(1 +
sizeOf(View)) if it doesn't have it already}
{It forwards the subscription to a node randomly chosen in its list if it does not keep it
keep=RandomChoiceBetween0and1 ()
keep=Math.Floor((view.Count+1)*keep);
if (keep==0) and s ∉ view then
   view.Add(s);
else
   int i=RandomChoice(view.Count);
   n=view[i];
   send(n,s,forwardedSusbcription);
end if
```

Note that our membership protocol creates a distribution graph which ensures that every node is connected. This implies that, in the absence of failures or unsubscriptions, the dissemination of messages is fully reliable.

Unsubscriptions Unsusbcriptions are handled as a gossip message and are disseminated to all members of the group. Any node that has the unsubscribing node in its partial view deletes it on receiving the unsubscription message.

Recovery from isolation A node becomes isolated when its identifier is present in no local views because, for example, all nodes holding its identifier have either failed or unsubscribed. Such a node has a substantial probability of remaining isolated for a long time. To overcome this problem, we propose a periodic check mechanism performed by

isolated nodes. A node which has not received messages for a given period (the period is chosen to be much larger than the average time between messages [1]) will resubscribe through an arbitrary node in its partial view.

3 Analysis

We now present the theoretical analysis of the algorithm described above. We model the system as a random directed graph: nodes correspond to subscribers and there is a directed arc (x,y) whenever y is in the local list of x[2].

When a new node subscribes, the action of our algorithm is to create a random number of additional arcs, as follows. Suppose there are n members already in the group. If the new node subscribes to a node with out-degree d, then $d+c+1$ arcs are added. The new node has out-degree 1, with list consisting of just the node it subscribed to. The node receiving the subscription forwards one copy of the node-id of the subscribing node to each of its neighbours, and an additional c copies to randomly chosen neighbours. These forwarded subscriptions may be kept by the neighbours or forwarded, but are not destroyed until some node keeps them. No node keeps multiple copies of the same subscription. In practice, each node chooses whether to keep a forwarded subscription with probability inversely proportional to the length of its current list. For ease of analysis, we'll assume that new arcs are added by choosing nodes uniformly at random without replacement.

Let M_n denote the number of arcs when the number of nodes has grown to n, so that the average out-degree of each node is M_n/n. We have

$$EM_n = \left(1+\frac{1}{n-1}\right) EM_{n-1} + c + 1,$$

from which we find that $EM_n \approx (c+1)n\log n$. If in fact $M_n = (c+1)n\log n$, and the arcs are distributed uniformly at random among the nodes, then it was shown in [8, Theorem 1] that the probability of a gossip being successful is very nearly 1 if the link failure probability is smaller than $c/(c+1)$. We shall now bound the deviation of the random quantity M_n from its mean, and show that, with high probability, M_n is very close to $(c+1)n\log n$. In other words, the proposed membership management scheme achieves the desired out-degree with high probability, with no centralized control or even knowledge of the size of the group.

Let $\mathcal{F}_n$ denote the σ-algebra corresponding to the sequence of random graphs created after each of the first n nodes joined the group. We shall show that

$$X_n := \frac{M_n}{n} - \sum_{i=1}^{n} \frac{c+1}{i}$$

[1] To facilitate this, we ensure that heartbeat messages are sent if no message has been sent within this period.

[2] Note that the graph represents the logical relation of membership in local views rather than the physical topology of the underlying network. The validity of the random graph model thus relies on the way in which the membership lists are created and is not dependent on the graph structure of the physical network.

is a martingale. By the assumption that new nodes subscribe to a randomly chosen member node, we have

$$E[M_{n+1}|\mathcal{F}_n] = M_n + \frac{M_n}{n} + c + 1,$$

from which it follows that $E[X_{n+1}|\mathcal{F}_n] = X_n$, i.e., X_n is a martingale. We now estimate its variance.

Let π_n denote the empirical distribution of node out-degrees conditional on $\mathcal{F}_n$. The subscription goes to a random node whose out-degree, denoted d_n, is a random draw from π_n. Now, $d_n + c$ copies of the subscription are forwarded, and are eventually kept by nodes chosen uniformly at random (without replacement)[3]. Let $d_1, \ldots, d_{d_n+c}$ denote the out-degrees of these nodes. The new empirical distribution is

$$\pi(n+1) = \frac{n}{n+1}\pi(n) + \frac{1}{n+1}\left(\delta_1 + \sum_{k=1}^{d_n+c}(\delta_{d_k+1} - \delta_{d_k})\right), \tag{1}$$

where δ_k denotes unit mass at k. Let f_n and v_n denote the expected mean and second moment of π_n, which is a random probability distribution. Let $h_n = E[d_i d_j]$ where d_i and d_j are the out-degrees of two distinct nodes chosen uniformly at random. Let w_n denote the expected second moment of the total number of edges, M_n.

Observe that $M_{n+1} = M_n + 1 + (d_n + c)$, and so $(n+1) f_{n+1} = n f_n + 1 + f_n + c$, i.e.,

$$f_{n+1} = f_n + (c+1)/(n+1). \tag{2}$$

Moreover,

$$\begin{aligned} w_{n+1} &= w_n + E[d_n^2 + 2(1+c)d_n + (1+c)^2] + 2(1+c)E[M_n] + 2E[d_n M_n] \\ &= w_n + v_n + 2(1+c) f_n + (1+c)^2 + 2(1+c)n f_n + 2\sum_{i=1}^{n} E[d_n d_i] \\ &= w_n + 3v_n + 2(n-1)h_n + 2(1+c)(n+1) f_n + (1+c)^2. \end{aligned}$$

We also have from (1) that

$$\begin{aligned} v_{n+1} &= \frac{1}{n+1} E\left(1 + \sum_{i=1}^{d_n+c}(d_i+1)^2 + \sum_{i=d_n+c+1}^{n} d_i^2\right) \\ &= \frac{1}{n+1}\left(1 + \sum_{i=1}^{n} E[d_i^2] + 2E\sum_{i=1}^{d_n+c} d_i + d_n + c\right) \\ &= \frac{1}{n+1}(1 + nv_n + 2E[d_1 d_n] + 2cE[d_n] + E[d_n] + c) \\ &= \frac{n}{n+1}v_n + \frac{2}{n+1}h_n + \frac{2c+1}{n+1} f_n + \frac{c+1}{n+1}. \end{aligned}$$

[3] In fact, our algorithm stores subscriptions preferentially in nodes with smaller out-degree. Ignoring this increases the variance of out-degrees and so the conclusions from the analysis presented here are expected to hold *a fortiori* for our algorithm.

We can eliminate h_n from the two equations above using the fact that

$$w_n = E[M_n^2] = E[(\sum_{i=1}^{n} d_i)^2] = nv_n + n(n-1)h_n,$$

from which it follows that

$$h_n = \frac{w_n - nv_n}{n(n-1)}.$$

Substituting this above and simplifying, we get the recursions:

$$w_{n+1} = \left(1+\frac{2}{n}\right) w_n + v_n + 2(1+c)(n+1) f_n + (1+c)^2 \tag{3}$$

$$v_{n+1} = \frac{n-2}{n-1} v_n + \frac{2}{(n-1)n(n+1)} w_n + \frac{2c+1}{n+1} f_n + \frac{c+1}{n+1}. \tag{4}$$

Let $\gamma_n = w_n - n^2 f_n^2$ denote the variance of M_n, and let $\eta_n = v_n - f_n^2$ denote the expected variance of the random distribution π_n. From the above, we obtain the following recursions for γ_n and η_n:

$$\gamma_{n+1} = \left(1+\frac{2}{n}\right) \gamma_n + \eta_n, \tag{5}$$

$$\eta_{n+1} = \frac{n-2}{n-1}\eta_n + \frac{2}{(n-1)n(n+1)}\gamma_n + \frac{1}{n+1}(f_n^2 - f_n) + \frac{c+1}{n+1} - \left(\frac{c+1}{n+1}\right)^2. \tag{6}$$

Iterating (5), we obtain the expression

$$\gamma_n = n(n+1) \sum_{k=0}^{n-1} \frac{\eta_k}{(k+1)(k+2)}. \tag{7}$$

We substitute this in (6) and use straightforward bounds to obtain the inequality

$$\eta_{n+1} \le \frac{n-2}{n-1}\eta_n + \frac{2}{n-1} \sum_{k=0}^{n-1} \frac{\eta_k}{(k+1)(k+2)} + \kappa \frac{\log^2 n}{n}, \tag{8}$$

valid for all $n \ge 2$, where κ is a suitably chosen constant (the expression for f_n entails for instance that $\kappa = 3(c+1)^2 + (c+1)/\log^2 2$ would suffice).

We now establish the following result.

Lemma 1. *There exists a constant $R > 0$ such that, for all $n \ge 2$,*

$$\eta_n \le R \log^2 n. \tag{9}$$

Proof: Assume that we have found a constant R such that the desired inequality is satisfied for all k in the range $\{2, \ldots, n\}$. In view of (8), we obtain

$$\eta_{n+1} \le R\log^2 n - \frac{R}{n-1}\log^2 n + \frac{2R}{n-1} \sum_{k \ge 2} \frac{\log^2 k}{(k+1)(k+2)} + \frac{\eta_0 + \eta_1}{n-1} + \kappa \frac{\log^2 n}{n}.$$

Splitting the second term into two halves, and introducing the notation

$$C = 2\sum_{k\geq 2} \frac{\log^2 k}{(k+1)(k+2)},$$

we obtain

$$\eta_{n+1} \leq R\log^2 n + (\kappa - R/2)\frac{\log^2 n}{n} + \frac{1}{n-1}(R(C - \frac{\log^2 n}{2}) + \eta_0 + \eta_1).$$

From this last equation, we see that the induction hypothesis carries over to $n+1$, provided the inequalities $R \geq 2\kappa$ and $R(C - \log^2 n/2) + \eta_0 + \eta_1 \leq 0$ hold. Let n_0 be the smallest index $k \geq 2$ such that $\log^2 k/2 - C > \eta_0 + \eta_1$.

We are now ready to choose the constant R. A suitable choice will be

$$R = \max\left(\max_{2\leq k\leq n_0}\left(\eta_k/\log^2 k\right), 2\kappa, 1\right).$$

Indeed, taking R larger than $\max_{2\leq k\leq n_0}(\eta_k/\log^2 k)$ ensures that the induction hypothesis is satisfied in the range $k = 2,\ldots,n_0$. Taking it larger than 2κ ensures that the first inequality we need to check in order to use induction is satisfied ; taking it larger than 1 ensures that, for $n \geq n_0$, the second inequality $R(c - \log^2 n) + \eta_0 + \eta_1 \leq 0$ is also satisfied, hence we can use induction from n_0 onwards.

Corollary 1. *There exists a constant $R' > 0$ such that, for all $n \geq 1$,*

$$\gamma_n \leq R' n^2. \tag{10}$$

Proof: Combining (7) and (9) yields

$$\gamma_n \leq 2n^2\left(\eta_0 + \eta_1 + R\sum_{k\geq 2}\frac{\log^2 k}{(k+1)(k+2)}\right),$$

from which the claim of the corollary follows if we choose $R' = 2(\eta_0 + \eta_1 + RC)$, where the constant C is as in the proof of the previous lemma.

We now obtain from this corollary that $\mathrm{Var}(X_n) = \mathrm{Var}(M_n)/n^2 \leq R'$ for all n. As a consequence, the martingale X_n is uniformly integrable, and by the martingale convergence theorem, it converges almost surely to a finite random variable X_∞ as $n \to \infty$. In other words, the mean out-degree M_n/n is close to the target value of $(c+1)\log n$ in the precise sense that their difference converges to a finite random variable (not growing with n) as $n \to \infty$. Thus, we finally obtain from Theorem 1 of [8] that gossip in the resulting random graph reaches all participants with high probability provided the proportion of failed links is smaller than $c/(c+1)$.

4 Simulation results

In this section we present some preliminary simulation results which confirm the theoretical analysis and show the self-organizing property of SCAMP as well as the good quality of the partial views generated. We first study the size of partial views and then provide some results comparing the resilience to failure of a gossip-based algorithm relying on SCAMP for membership management with one relying on a global scheme.

4.1 View size

The first objective of SCAMP is to ensure that each node has a randomized partial view of the membership, of the right size to ensure successful gossip. All experiments in this section have been done with $c = 0$, i.e., the objective is to achieve an average view size of $\log(n)$. Recall that a fanout of this order is required to ensure that gossip is successful with high probability. The key result we want to confirm here is that a fully decentralized scheme as in SCAMP can provide each node with a partial view of size approximately $\log(n)$, without global membership information or synchronization between nodes.

In Figure 1, we plot the average size of partial views achieved by SCAMP against system size. The figure shows that the average list size achieved by SCAMP matches the target value very closely, supporting our claim that SCAMP is self-organizing. Figure 2 shows the distribution of list sizes of individual nodes in a 5000 node system. The distribution is unimodal with mode approximately at $\log(n)$ ($\log(5000) = 8.51$). While analytical results on the success probability of gossip were derived in [8] for two specific list size distributions, namely the deterministic and binomial distributions, we believe that the results are largely insensitive to the actual degree distribution and depend primarily on the mean degree.[4] This is corroborated by simulations.

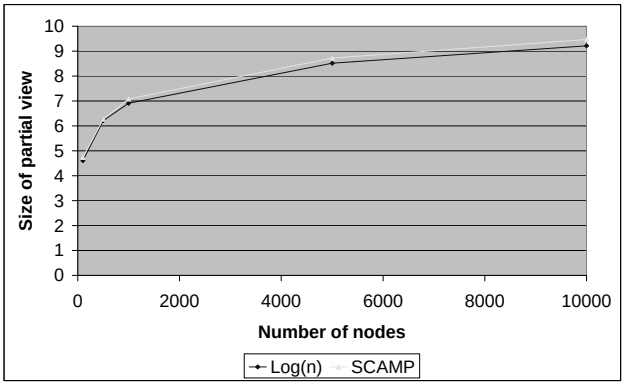

Fig. 1. Relation between system size and average list size produced by SCAMP

4.2 Resilience to failures

One of the most attractive features of gossip-based multicast is its robustness to node and link failures. Event dissemination can meet stringent reliability guarantees in the

[4] The claim has to be qualified somewhat as the following counterexample shows. If the fanout is n with probability $\log n/n$ and zero with probability $1 - (\log n/n)$, then the mean fanout is $\log n$ but the success probability is close to zero. Barring such extremely skewed distributions, we believe the claim to be true. An open problem is to state and prove a suitable version of this claim.

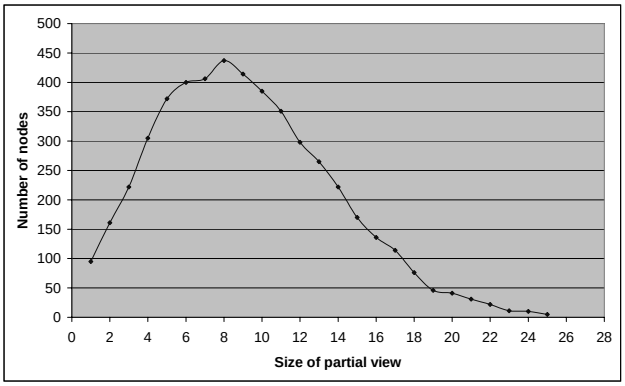

Fig. 2. Histogram of list sizes at individual nodes in a 5000 node system

presence of failures, without any explicit recovery mechanism. This makes these protocols particularly attractive in highly dynamic environements where members can disconnect for non-negligible periods and then reconnect.

We compare a gossip-based protocol relying on SCAMP with one relying on global knowledge of membership in terms of their resilience to node failures. Figure 3 depicts the simulation results. We plot the fraction of surviving nodes reached by a gossip message initiated from a random node as a function of the number of failed nodes. Two observations are notable. First, the fraction of nodes reached remains very high even when close to half the nodes have failed, which confirms the remarkable fault-tolerance of gossip-based schemes. Second, this fraction is almost as high using SCAMP as using a scheme requiring global knowledge of membership. This attests to the quality of the partial views provided by SCAMP and demonstrates its viability as a membership scheme for supporting gossip.

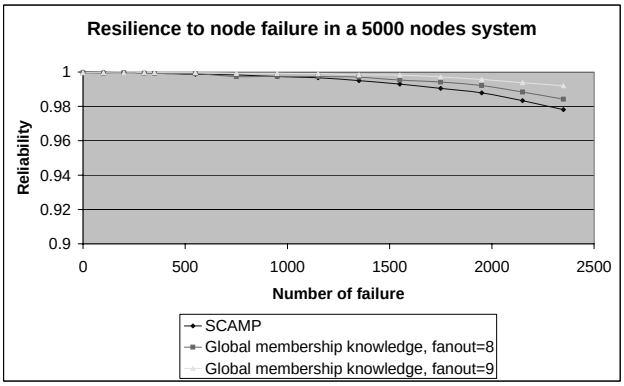

Fig. 3. Resilience to failure in a system of 5000 node system

5 Conclusion

Reliable group communication is important in applications involving large-scale distributed systems. Probabilistic gossip-based protocols have proven to scale to a large number of nodes while providing attractive reliability properties. However, most gossip-based protocols rely on nodes having global membership information. For large groups, this consumes a lot of memory and generates a lot of network traffic due to the synchronization required to maintain global consistency. In order to use gossip-based algorithms in large-scale groups, which is their natural application domain, the membership protocol also needs to be decentralized and lightweight.

In this paper, we have presented the design, theoretical analysis and evaluation of SCAMP, a probabilistic peer-to-peer scalable membership protocol for gossip-based dissemination. SCAMP is fully decentralized in the sense that each node maintains only a partial view of the system. It is also self-organizing: the size of partial views naturally increases with the number of subscriptions in order to ensure the same reliability guarantees as the group grows. Thus SCAMP provides efficient support for large and highly dynamic groups.

One of the key contributions of this paper is the theoretical analysis of SCAMP, which establishes probabilistic guarantees on its performance. The analysis, which is asymptotic, is confirmed by simulations, which show that a gossip-based protocol using SCAMP as a membership service is almost as resilient to failures as a protocol relying on knowledge of global membership at each node.

Future work includes comparing SCAMP with other membership protocols, and modifying it to take geographical locality into account in the generation of partial views.

References

1. T. Anker, G.V. Chockler, D. Dolev, and I. Keidar. Scalable group membership services for novel applications. In Michael Merrit M. Mavronicolas and Nir Shavit, editors, *Networks in Distributing Computing (DIMACS workshop)*, DIMACS 45, pages 23–42. American Mathematical Society, 1998.
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. S. Deering and D. Cheriton. Multicast Routing in Datagram Internetworks and Extended LANs. *ACM Transactions on Computer Systems*, 8(2), May 1990.
4. S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. The PIM Architecture for Wide-Area Multicast Routing. *IEEE/ACM Transactions on Networking*, 4(2), April 1996.
5. P.T. Eugster, R. Guerraoui, S.B. Handurukande, A.-M. Kermarrec, and P. Kouznetsov. Lightweight probabilistic broadcast. In *IEEE International Conference on Dependable Systems and Networks (DSN2001*, 2001.
6. 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 Transaction on networking*, pages 784–803, December 1997.
7. I. Keidar, J. Sussman, K. Marzullo, and D. Dolev. A client-server oriented algorithm for virtually synchronous group memebership in wan's. In *20th International Conference on Distributed Computing Systems (ICDCS)*, pages 356–365, April 2000.

8. A.-M Kermarrec, L. Massoulié, and A.J. Ganesh. Probabilistic reliable dissemination in large-scale systems. Submitted for publication (available at http://research.microsoft.com/camdis/gossip.htm).
9. J.C. Lin and S. Paul. A reliable multicast transport protocol. In *Proc. of IEEE INFOCOM'96*, pages 1414–1424, 1996.
10. 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.
11. M.-J. Lin, K. Marzullo, and S. Masini. Gossip versus deterministic flooding: Low message overhead and high-reliability for broadcasting on small networks. In *Proceedings of 14th International Symposium on DIStributed Computing (DISC 2000*, pages 253–267, Toledo, Spain, October 4-6 2000.
12. 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.