

The background of the page features a large, faint, golden seal of the University of Bologna. The seal is circular and contains a central figure, likely a saint or historical figure, surrounded by Latin text. The text includes "UNIVERSITAS BOLOGNENSIS" at the top, "S. PETRI UBIQUE PATRIS" on the right, and "SIGILLUM" at the bottom. The seal is partially obscured by the text of the report.

Gossip-based Unstructured Overlay Networks: An Experimental Evaluation

Márk Jelasity

Rachid Guerraoui

Anne-Marie Kermarrec

Maarten van Steen

Technical Report UBLCS-2003-15

December 2003

Department of Computer Science

University of Bologna

Mura Anteo Zamboni 7
40127 Bologna (Italy)

The University of Bologna Department of Computer Science Research Technical Reports are available in gzipped PostScript format via anonymous FTP from the area `ftp.cs.unibo.it:/pub/TR/UBLCS` or via WWW at URL `http://www.cs.unibo.it/`. Plain-text abstracts organized by year are available in the directory ABSTRACTS. All local authors can be reached via e-mail at the address `last-name@cs.unibo.it`.

Recent Titles from the UBLCS Technical Report Series

- 2002-8 *User Untraceability in the Next-Generation Internet: a Proposal*, Tortonesi, M., Davoli, R., August 2002.
- 2002-9 *Towards Adaptive, Resilient and Self-Organizing Peer-to-Peer Systems*, Montresor, A., Meling, H., Babaoglu, O., September 2002.
- 2002-10 *Towards Self-Organizing, Self-Repairing and Resilient Distributed Systems*, Montresor, A., Babaoglu, O., Meling, H., September 2002 (Revised November 2002).
- 2002-11 *Messor: Load-Balancing through a Swarm of Autonomous Agents*, Montresor, A., Meling, H., Babaoglu, O., September 2002.
- 2002-12 *Johanna: Open Collaborative Technologies for Teleorganizations*, Gaspari, M., Picci, L., Petrucci, A., Faglioni, G., December 2002.
- 2003-1 *Security and Performance Analyses in Distributed Systems (Ph.D Thesis)*, Aldini, A., February 2003.
- 2003-2 *Models and Types for Wide Area Computing. The calculus of Boxed Ambients (Ph.D. Thesis)*, Crafa, S., February 2003.
- 2003-3 *MathML Formatting (Ph.D. Thesis)*, Padovani, L., February 2003.
- 2003-4 *Performance Evaluation of Mobile Agents Paradigm for Wireless Networks (Ph.D. Thesis)*, Al Mobaideen, W., March 2003.
- 2003-5 *Synchronized Hypermedia Documents: a Model and its Applications (Ph.D. Thesis)*, Gaggi, O., March 2003.
- 2003-6 *Searching and Retrieving in Content-Based Repositories of Formal Mathematical Knowledge (Ph.D. Thesis)*, Guidi, F., March 2003.
- 2003-7 *Intersection Types, Lambda Abstraction Algebras and Lambda Theories (Ph.D. Thesis)*, Lusin, S., March 2003.
- 2003-8 *Towards an Ontology-Guided Search Engine*, Gaspari, M., Guidi, D., June 2003.
- 2003-9 *An Object Based Algebra for Specifying A Fault Tolerant Software Architecture*, Dragoni, N., Gaspari, M., June 2003.
- 2003-10 *A Scalable Architecture for Responsive Auction Services Over the Internet*, Amoroso, A., Fanzieri F., June 2003.
- 2003-11 *WSSecSpaces: a Secure Data-Driven Coordination Service for Web Services Applications*, Lucchi, R., Zavattaro, G., September 2003.
- 2003-12 *Integrating Agent Communication Languages in Open Services Architectures*, Dragoni, N., Gaspari, M., October 2003.
- 2003-13 *Perfect load balancing on anonymous trees*, Margara, L., Pistocchi, A., Vassura, M., October 2003.
- 2003-14 *Towards Secure Epidemics: Detection and Removal of Malicious Peers in Epidemic-Style Protocols*, Jelasity, M., Montresor, A., Babaoglu, O., November 2003.
- 2003-15 *Gossip-based Unstructured Overlay Networks: An Experimental Evaluation*, Jelasity, M., Guerraoui, R., Kermarrec, A-M., van Steen, M., December 2003.
- 2003-16 *Robust Aggregation Protocols for Large-Scale Overlay Networks*, Jelasity, M., Montresor, A., Babaoglu, O., December 2003.

Gossip-based Unstructured Overlay Networks: An Experimental Evaluation¹

Márk Jelasity² Rachid Guerraoui³ Anne-Marie Kermarrec⁴ Maarten van Steen⁵

Technical Report UBLCS-2003-15

December 2003

Abstract

Gossip-based protocols offer a scalable and reliable approach to a number of large-scale distributed applications. The basic idea is for each node to periodically select a random peer node to exchange information with. Analytical studies reveal a high reliability of gossip-based protocols. However, a usual assumption of these studies is that the peer is chosen uniformly at random from the set of all nodes. In practice—instead of requiring all nodes to know all the peer nodes—a scalable way to implement random peer selection is by constructing dynamic unstructured overlays through gossiping membership information itself. In this paper we generalize existing gossip-based overlays by introducing a general scheme in which existing overlays as well as novel protocols can be implemented. The central theme of this paper is exploring and comparing several implementations of our abstract scheme. Through extensive experimental analysis, we show that all of them lead to different communication topologies none of which is uniformly random. This clearly renders traditional theoretical approaches invalid. Our observations help explain important differences between design choices of a gossip-based protocol and how these impact the reliability of the overlay. Understanding these differences poses new interesting theoretical problems.

1. This work was partially supported by the Future & Emerging Technologies unit of the European Commission through Project BISON (IST-2001-38923).
2. University of Bologna, Italy
3. EPFL, Lausanne, Switzerland
4. Microsoft Research, Cambridge, UK
5. Vrije Universiteit, Amsterdam, The Netherlands

1 Introduction

Motivation Recently, a number of simple, scalable, fully distributed gossip-based protocols have been proposed for solving various problems including aggregation [14, 13], information dissemination [7, 5], load balancing [15], and topology management [26]. The common property of these protocols is that periodically, every node of the distributed system exchanges information with some of its peers, which are chosen in a random manner. In all cases, many desirable features like scalability, reliability, and efficiency have been rigorously analyzed (see, e.g., for information dissemination [20]). However, theoretical analysis normally assumes that the peers that any given node selects and exchanges information with represent a uniform random sample of *all nodes* in the system.

To achieve this uniform random selection, many gossip-based protocols assume that every node *knows* all other nodes of the system [4, 11, 16]. Practically speaking, every node has to maintain a membership table, also called its *view*, the size of which grows with the size of the system. The cost of maintaining such tables has a non-negligible overhead in a dynamic system where processes join and leave at run time. In short, whereas the information dissemination is scalable, the underlying membership protocol is not.

Recently, to solve this problem, much research has been devoted to designing scalable membership protocols to build an overlay network connecting, in a decentralized and scalable way, a large set of nodes. The basic idea is to use a gossip-based dissemination of membership information itself [6]. The continuous gossiping of membership information enables the building of unstructured overlay networks that capture the dynamic nature of distributed peer-to-peer systems and help provide very good connectivity in the presence of failures or peer disconnections. Interestingly, there are many variants of the basic gossip-based membership dissemination idea, and these variants mainly differ in the way new views are built after merging and truncating views of communicating peers (see, e.g., [12]). So far, however, there has never been any evaluation and comparison between these variants, and this makes it hard for a programmer to choose the protocol that best suits the application needs. More importantly, it is not clear whether any of these variants actually lead to *uniform sampling*, which lies at the heart of all analytical studies of the effects of gossip-based protocols, such as the reliability of dissemination. In search for an answer to these questions, this paper introduces a generic protocol scheme in which known and novel gossip-based unstructured overlay protocols can be instantiated, and presents an extensive empirical comparison of these protocols.

Contribution First, we present a generic protocol scheme, which generalizes the gossip-based overlay protocols we are aware of, and which makes it possible to implement new protocols as well. In other words, we give a generic gossip-based framework that provides the means to implement various decentralized membership *policies*, which implicitly define unstructured overlay networks. Secondly, we describe an experimental methodology to evaluate the *communication topologies* of the resulting overlay networks. In particular, we examine if the overlays exhibit *stable properties*, that is, whether the corresponding protocol instances lead to the *convergence* of important properties of the overlay. Thirdly, we measure the extent to which these communication topologies deviate from the desirable uniform random model mentioned earlier. We do so by looking at several static and dynamic properties: degree distribution, average path length, clustering coefficient, and self-healing capacity.

The behavior of the protocol instances we evaluate shows a rather wide variation. A common characteristic, however, is that no instance leads to a uniform sampling, rendering traditional theoretical approaches invalid. This result is somewhat surprising, and if there is one important lesson to learn from our experiments, it is that the emergent behavior exhibited by various gossip-based protocols may not be that easy to capture in mathematical models. Nevertheless, our experimental results reveal some discrepancies of the different protocol instances in varying scenarios with respect to various topological properties and self-healing capabilities.

Roadmap Section 2 describes our generic protocol and the various dimensions according to which it can be instantiated. Section 3 presents our experimentation methodology. Sections 4, 5 and 6 discuss our results in different simulation scenarios. In Section 7 we interpret the result of the experiments. Related work is discussed in Section 8. Finally, Section 9 concludes the paper.

2 Evaluation Framework

To study the impact on various parameters of gossip-based approaches to maintain unstructured overlays, we define an evaluation framework. A wide range of protocols fits into this framework and in particular the protocols Lpbcast [6] and Newscast [12] are specific instances of protocols within this framework.

```

do forever
  wait(T time units)
   $p \leftarrow \text{selectPeer}()$ 
  if push then
    // 0 is the initial hop count
    myDescriptor  $\leftarrow$  (myAddress, 0)
    buffer  $\leftarrow$  merge(view, {myDescriptor})
    send buffer to  $p$ 
  if pull then
    receive view $p$  from  $p$ 
    view $p$   $\leftarrow$  increaseHopCount(view $p$ )
    buffer  $\leftarrow$  merge(view $p$ , view)
    view  $\leftarrow$  selectView(buffer)

```

(a) active thread

```

do forever
  ( $p, \text{view}_p$ )  $\leftarrow$  waitMessage()
  view $p$   $\leftarrow$  increaseHopCount(view $p$ )
  if pull then
    // 0 is the initial hop count
    myDescriptor  $\leftarrow$  (myAddress, 0)
    buffer  $\leftarrow$  merge(view, {myDescriptor})
    send buffer to  $p$ 
  buffer  $\leftarrow$  merge(view $p$ , view)
  view  $\leftarrow$  selectView(buffer)

```

(b) passive thread

Figure 1. The skeleton of a gossip-based protocol for maintaining unstructured overlay networks.

System model We consider a set of nodes connected in a network. A node has an address that is needed for sending a message to that node. Each node maintains addresses by means of a *partial view*, which is a set of c *node descriptors*. The value of c is the same for all nodes. Besides an address, a node descriptor also contains a *hop count*, as we explain below.

We assume that each node executes the same protocol, of which the skeleton is shown in Figure 1. The protocol consists of two threads: an active thread initiating communication with other nodes, and a passive thread waiting for incoming messages. The skeleton code is parameterized with two Booleans (push and pull), and two function placeholders (`selectPeer()` and `selectView()`).

A view is organized as a list with at most one descriptor per node and ordered according to increasing hop count. We can thus meaningfully refer to the *first* or *last* k elements of a view. A call to `increaseHopCount(view)` increments the hop count of every element in *view*. A call to `merge(view1, view2)` returns the union of *view₁* and *view₂*, ordered again by hop count. When there is a descriptor for the same node in each view, only the one with the lowest hop count is inserted into the merged view; the other is discarded.

This design space enables us to evaluate in a simple and rigorous way the impact of the various parameters involved in gossip-based protocols along three dimensions: (i) Peer selection; (ii) View propagation; (iii) View selection. Many variations exist along each of these dimensions; we limit our study to the three most relevant strategies per dimension. We shall now define these dimensions.

Peer selection Periodically, each node selects a peer to and exchange membership information with. This selection is implemented by the function `selectPeer()` that returns the address of a *live* node as found in the caller's current view. In this study, we consider the following *peer selection* policies:

| | |
|-------------|--|
| rand | Uniform randomly select an available node from the view |
| head | Select the first available node from the view (i.e., the one with the <i>lowest</i> hop count) |
| tail | Select the last available node from the view (i.e., the one with the <i>highest</i> hop count) |

View propagation Once a peer has been chosen, the peers may exchange information in various ways. We consider the following three *view propagation* policies:

| | |
|-----------------|--|
| push | The node sends its view to the selected peer |
| pull | The node requests the view from the selected peer |
| pushpull | The node and selected peer exchange their respective views |

View selection Once membership information has been exchanged between peers and merged as explained above, peers may need to truncate their views in order to adhere to the c items limit imposed as a protocol parameter. The function $\text{selectView}(\text{view})$ selects a subset of at most c elements from view . Again, we consider only three out of the many possible *view selection* policies:

| | |
|-------------|---|
| rand | Uniform randomly select c elements from view |
| head | Select the first c elements from view |
| tail | Select the last c elements from view |

These three types of policies give rise to a total of 27 combinations, each of which we express by means of a 3-tuple (ps, vs, vp) with ps indicating one of the three possible peer selection policies, vs the view selection policies, and vp the chosen view propagation policy. As an example, Lpbcast corresponds to the 3-tuple (rand,rand,push), whereas Newscast is described by (rand,head,pushpull). In the following, a DON'T CARE value (i.e., a wild card) is denoted by the symbol “*”.

3 Experimental methodology

As we explained in Section 1 our experiments focus on the *communication topology* or *overlay topology* defined by the set of nodes and their views. From now on we adopt a graph theoretic terminology when referring to this topology. In this framework the directed edges of the communication graph are defined as follows. If node a stores the descriptor of node b in its view then there is a directed edge (a, b) from a to b .

3.1 Targeted questions

There are two general questions we seek to answer. The first and most fundamental question is whether, for a particular protocol implementation, the communication graph has some stable properties, which it maintains during the execution of the protocol. In other words, we are interested in the *convergence behavior* of the protocols. We can expect several sorts of dynamics which include chaotic behavior, oscillations or convergence. In case of convergence the resulting state may or may not depend on the initial configuration of the system. In the case of overlay networks we prefer to have convergence toward a state that is independent of the initial configuration. Sometimes this property is called *self-organization*. In our case it is essential that in a wide range of scenarios the system should automatically produce consistent and predictable behavior.

Another related question is in order: *if* there is convergence then what kind of communication graph the protocol converges to? In particular, as mentioned earlier, we are interested in what sense do these graphs deviate from certain random graph models.

3.2 Selected graph properties

In order to find answers to the above problems we need to select a set of observable properties that characterize the communication graph. In the following, we will focus on the *undirected* version of the communication graph which we get by simply dropping the orientation of the edges. The reason for this choice is that even if the “knows-about” relation that defines the directed communication graph is one-way, the actual information flow from the point of view of the applications of the overlay is potentially two-way, since after initiating a connection the passive party will learn about the active party as well. Nevertheless, when it is appropriate, we will comment on the directed version of the graph. Now let us turn to the properties we will examine.

Degree distribution The degree of a node is defined as the number of its neighbors in the undirected communication graph. We will consider several aspects of the degree distribution including average degree, the dynamics of the degree of a node, and the exact degree distribution. The motivations for looking at degree distribution is threefold and includes its direct relationship with reliability to different patterns of node failures [2]; its crucial effect on the exact way epidemics are spread (and therefore on the way epidemic-based broadcasting is performed) [19]; and finally its key role in determining if there are communication hot spots in the overlay.

| protocol | part.-ed runs | avg. num. of clusters | avg. largest cluster |
|------------------|---------------|-----------------------|----------------------|
| (rand,head,push) | 100% | 58.36 | 4112.09 |
| (rand,rand,push) | 33% | 2.27 | 9572.18 |
| (tail,head,push) | 100% | 38.19 | 7150.52 |
| (tail,rand,push) | 1% | 2.00 | 9941.00 |

Table 1. Protocols where partitioning was observed in the growing overlay scenario. Data corresponds to cycle 300.

Average path length The shortest path length between node a and b is the minimal number of edges that are necessary to traverse in order to reach b from a in the graph. The average path length is the average of shortest path lengths over all pairs of nodes in the graph. The motivation of looking at this property is that, in any information dissemination scenario, the shortest path length defines a lower bound on the time and costs of reaching a peer. For scalability small average path length is essential.

Clustering coefficient The clustering coefficient of a node a is defined as the number of edges between the neighbors of a divided by the number of all possible edges between those neighbors. Intuitively, this coefficient indicates the extent to which the neighbors of a are also neighbors of each other. The clustering coefficient of the graph is the average of the clustering coefficients of the nodes, and always lies between 0 and 1. For a complete graph, it is 1, for a tree it is 0. The motivation for analyzing this property is that a high clustering coefficient has potentially damaging effect on both information dissemination (by increasing the number of redundant messages) and also on the self-healing capacity by weakening the connection of a cluster to the rest of the graph thereby increasing the probability of partitioning. Furthermore, it provides an interesting possibility to draw parallels with research on complex networks where clustering is an important research topic (e.g., in social networks) [27].

3.3 Parameter settings

The main goal of this paper is to explore the different design choices in the protocol space described in Section 2. That is, the parameters which we want to explore are peer selection, view selection, and symmetry model. Accordingly, we chose to fix the network size to $N = 10^4$ and the maximal view size to $c = 30$.

During our preliminary experiments some parameter settings turned out not to result in meaningful overlay management protocols. In particular, (head,*,*) results in severe clustering, (*,tail,*) cannot handle dynamism (joining nodes) at all and (*,*,pull) converges to a star topology, which is highly undesirable. These variants are therefore excluded from further discussion.

4 Convergence

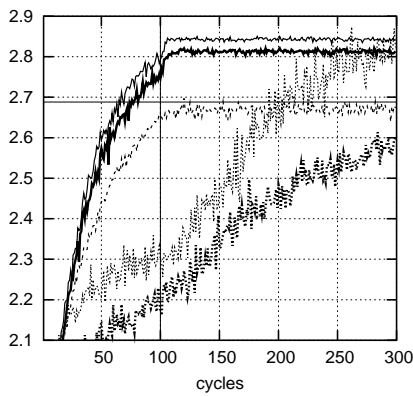
We now present experimental results that illustrate the convergence properties of the protocols in different bootstrapping scenarios. This section describes the motivation and exact design of the experiments and presents the output. Interpretation and discussion of the results is postponed until Section 7.

We will consider three scenarios to test convergence. The first is the case of a growing overlay discussed in Section 4.1. The second is the initialization of the overlay with a structured large diameter topology (Section 4.2) and finally the initialization with a random topology (Section 4.3).

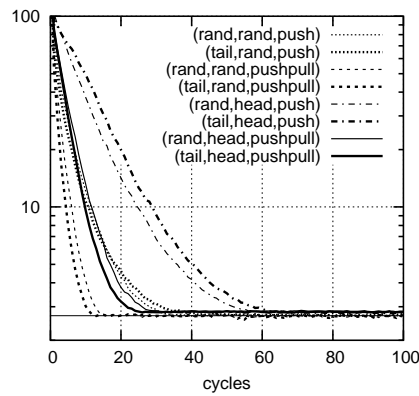
As we focus on the dynamical properties of the protocols, we did not wish to average out interesting patterns so in all cases the result of a single run is shown in the plots. Nevertheless, we ran all the scenarios 100 times to gain data on the stability of the protocols with respect to the connectivity of the overlay. Connectivity is a crucial feature, a minimal requirement for all applications. The results of these runs show that in all scenarios, every protocol under examination creates a connected overlay network in 100% of the runs. The only exceptions (shown in Table 1) were detected during the growing overlay scenario.

4.1 Growing overlay

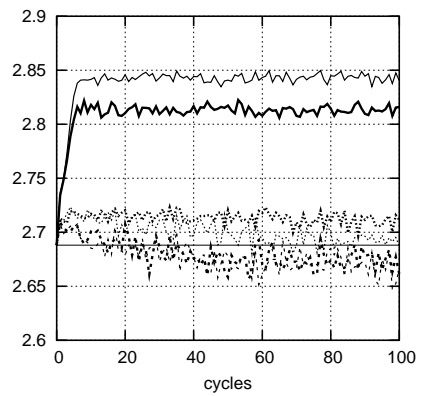
In this scenario the overlay network initially contains only one node. At the beginning of each cycle, 100 new nodes are added to the network until the maximal size is reached in cycle 100. The view of these nodes is initialized with only a single



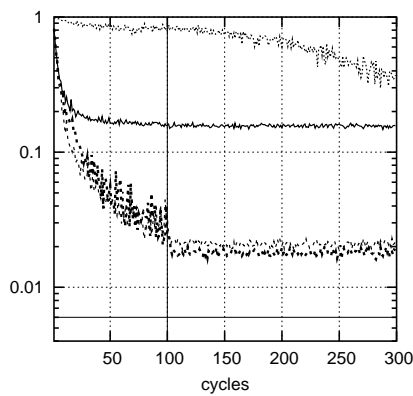
(a) growing, average path length



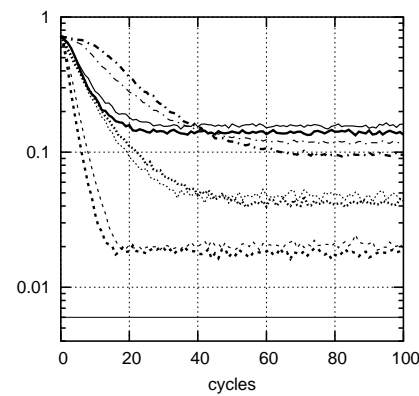
(b) lattice, average path length



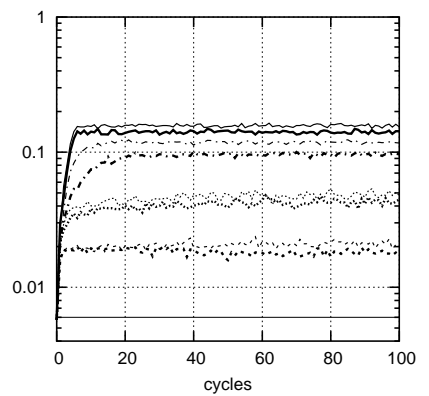
(c) random, average path length



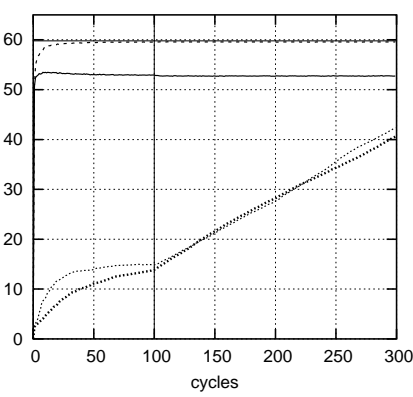
(d) growing, clustering coefficient



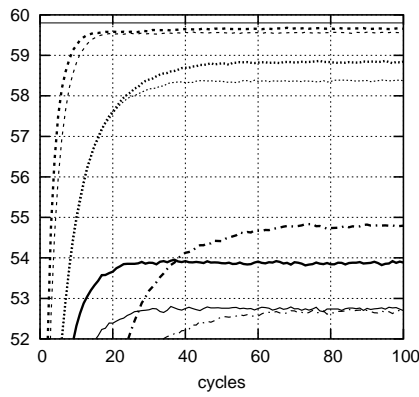
(e) lattice, clustering coefficient



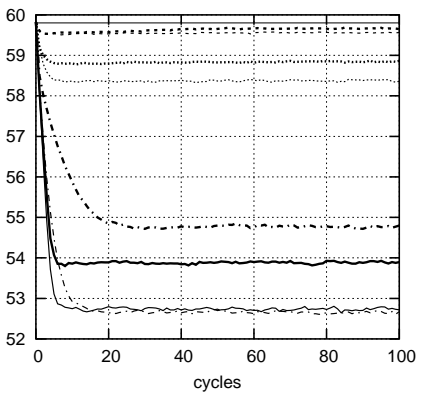
(f) random, clustering coefficient



(g) growing, average node degree



(h) lattice, average node degree



(i) random, average node degree

Figure 2. Dynamics of graph properties. Horizontal line indicates the property in a uniform random topology, vertical line indicates end of growth.

node descriptor, which belongs to the oldest, initial node.

This scenario is the most pessimistic one for bootstrapping the overlays. It would be straightforward to improve it by using more contact nodes, which can come from a fixed list or which can be obtained using inexpensive local random walks on the existing overlay. However, in our discussion we intentionally avoid such optimizations to allow a better focus on the core protocols and their differences.

Figure 2 shows the dynamics of the properties of the communication topology. For clarity reasons, in some plots we have not shown (tail,*,*) versions of some protocols that exhibit similar behavior to their respective (rand,*,*) version (but all protocols are shown in the other scenarios).

Protocols (rand,head,push) and (tail,head,push) are not plotted either due to their instability in this scenario with respect to connectivity of the overlay (see Table 1). A nonpartitioned run of both (rand,rand,push) and (tail,rand,push) is included however.

4.2 Ring lattice initial topology

In this scenario, the initial topology of the overlay was a ring lattice, a structured topology. The motivation behind this experiment is to examine if the overlay properties converge to the same random structure with a low average path length even if the initial topology is highly structured and has a large average path length.

We build the ring lattice as follows. The nodes are first connected into a ring in which each node has a descriptor in its view that belongs to its two neighbors in the ring. Subsequently, for each node, we add additional descriptors of the nearest nodes in the ring until the view is filled.

Figure 2 shows the output of this scenario as well. As in the case of the growing scenario, 300 cycles were run but only 100 are shown to focus on the more interesting initial dynamics of the protocols.

4.3 Random initial topology

In this scenario the initial topology was defined by a random graph, in which the views of the nodes were initialized by a uniform random sample of the peer nodes. Figure 2 includes the output of this scenario as well. As in the other scenarios, 300 cycles were run but only 100 are shown.

5 Degree distribution

The results presented in this section were obtained from the experiments performed according to the random initialization scenario described above. Like in the previous section, interpretation is postponed until Section 7.

When describing degree distribution in a dynamic system one has to focus on two aspects: the dynamics of the degree of individual nodes and the dynamics of the degree distribution over the whole overlay. In principle, knowing one of these aspects will not determine the other, and both are important properties of an overlay.

The evolution of the degree distribution over the whole overlay is shown in Figure 3. We can observe how the distribution reaches its final shape starting from the random topology, as the distributions that correspond to exponentially increasing time intervals (cycle 0, 3, 30 and 300) are also shown.

Let us continue with the question whether the distribution of the degree of a fixed node over time is the same as the distribution of the converged overlay at a fixed cycle. In the overlay the degree of 50 nodes were traced during $K = 300$ cycles. Table 2 shows statistical data concerning degree distribution over time at the 50 fixed nodes and over the full overlay in the last cycle (i.e. in cycle K). The notations used are as follows. Let $d(i, j)$ denote the degree of node i in cycle j . Let \bar{d}_i be the mean degree of node i over K consecutive cycles. Now, let $\bar{d} = \sum_{i=1}^{50} \bar{d}_i / 50$ and $\sigma = \sum_{i=1}^{50} (\bar{d}_i - \bar{d})^2 / 49$, where \bar{d} is the average and σ is the empirical variance of the time-averages of the degree of the traced 50 nodes. Finally, $\overline{D_K}$ is the average of node degrees in cycle K over all nodes.

The last question we consider is whether the sequence of node degrees during the cycles of the protocol can be considered a random sequence drawn from the overall degree distribution. If not, then how quickly does it change, and is it perhaps periodical? To this end we present autocorrelation data of the degree time-series of fixed nodes in Figure 4. Band indicates 99% confidence interval assuming the data is random. The autocorrelation of the series $d(i, 1), \dots, d(i, K)$ for a given time lag k is defined as

$$r_k = \frac{\sum_{j=1}^{K-k} (d(i, j) - \bar{d}_i)(d(i, j+k) - \bar{d}_i)}{\sum_{j=1}^K (d(i, j) - \bar{d}_i)^2},$$

which expresses the correlation of pairs of degree values separated by k cycles.

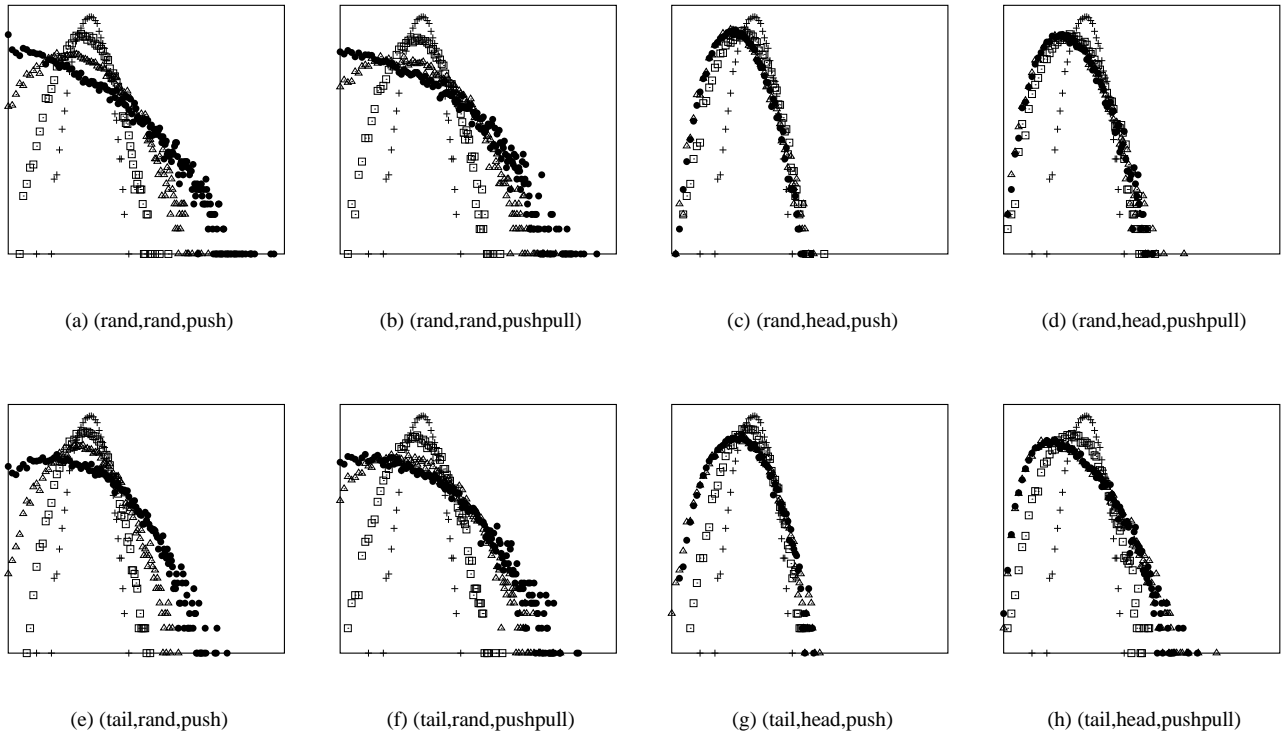


Figure 3. Degree distributions on the log-log scale, when starting from a random topology. The ranges are $[30,300]$ for the degree axis (horizontal), and $[1:1000]$ for the frequency axis (vertical). The symbol $+$ denotes the random graph (cycle 0). Empty box, empty triangle and filled circle belong to cycle 3, 30 and 300, respectively.

| protocol | \overline{D}_{300} | \bar{d} | $\sqrt{\sigma}$ |
|----------------------|----------------------|-----------|-----------------|
| (rand,head,push) | 52.623 | 52.703 | 1.394 |
| (tail,head,push) | 54.785 | 55.519 | 2.690 |
| (rand,head,pushpull) | 52.717 | 52.933 | 1.756 |
| (tail,head,pushpull) | 53.916 | 53.888 | 2.176 |
| (rand,rand,push) | 58.404 | 60.804 | 19.062 |
| (tail,rand,push) | 58.844 | 58.746 | 17.287 |
| (rand,rand,pushpull) | 59.569 | 61.306 | 13.886 |
| (tail,rand,pushpull) | 59.666 | 58.616 | 9.756 |

Table 2. Statistics describing the dynamics of the degree of individual nodes.

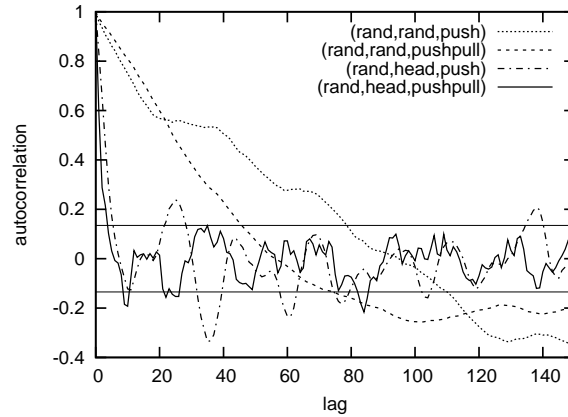


Figure 4. Autocorrelation of the degree of a fixed random node as a function of time lag, measured in cycles, computed from a 300 cycle sample. Protocols (tail,*,*) are omitted for clarity.

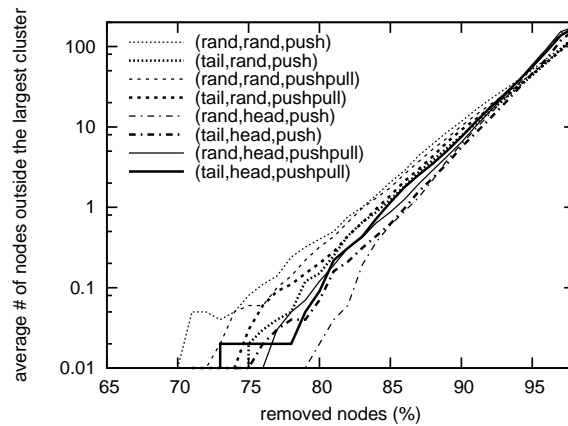


Figure 5. The number of nodes that do not belong to the largest connected cluster. The average of 100 experiments is shown.

For the correct interpretation of the figure observe that (rand,head,pushpull) can be considered practically random according to the 99% confidence band, while the time series produced by (rand,head,push) shows some weak high frequency periodic behavior. The protocols (*,rand,*) appear to show low frequency periodic behavior with strong short-term correlation, although to confirm that further experiments are necessary.

6 Self-healing capacity

As in the case of the degree distribution, the response of the protocols to a massive failure has a static and a dynamic aspect. In the static setting we are interested in the self-healing capacity of the converged overlays to a (potentially massive) node failure, as a function of the number of failing nodes. Removing a large number of nodes will inevitably cause some serious structural changes in the overlay even if otherwise it remains “usable,” that is, at least connected. In the dynamic case we would like to learn if and how the protocols can repair the overlay after a severe damage.

The effect of a massive node failure on connectivity is shown in Figure 5. In this setting the overlay in cycle 300 of the random initialization scenario was used as converged topology. From this topology, random nodes were removed and the connectivity of the remaining nodes was analyzed. In all of the $100 \times 8 = 800$ experiments performed we did not observe partitioning until removing 69% of the nodes. The figure depicts the number of the nodes outside the largest connected cluster. We observe consistent partitioning behavior over all protocol instances: even when partitioning occurs, most of the

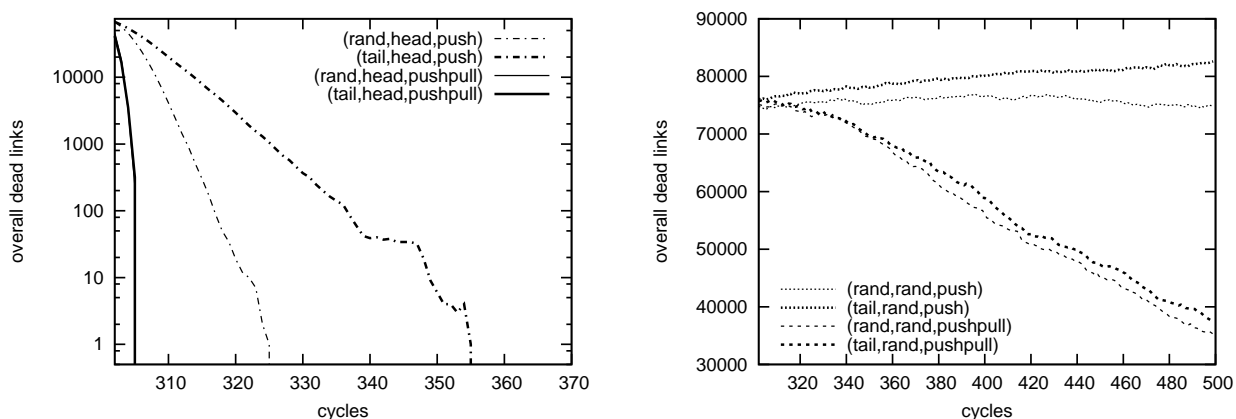


Figure 6. Number of dead links in the overlay after failure of 50% of the nodes. The $(*,head,pushpull)$ protocols fully overlap. Note the different scales of the two plots.

nodes form a single large connected cluster. Note that this phenomenon is well known for traditional random graphs [18].

In the dynamic scenario we made 50% of the nodes fail in cycle 300 of the random initialization scenario and we then continued running the protocols on the damaged overlay. The damage is expressed by the fact that, on average, half of the view of each node consists of descriptors that belong to nodes that are no longer in the network. We will call these descriptors dead links. Figure 6 shows how fast the protocols repair the overlay, that is, remove dead links. Based on the static node failure experiment it was expected that the remaining 50% of the overlay is not partitioned and indeed, we did not observe partitioning with any of the protocols.

7 Discussion

In our analysis of the output of the experiments presented above we first concentrate of the two main questions we posed: convergence and randomness. Then we move on to discuss the effects of the design choices in the three dimensions of the protocol space: peer selection, view selection, and symmetry of communication.

7.1 Convergence

Figures 2(d), 2(e) and 2(f) illustrate especially well that the protocols converge to the same clustering coefficient from extremely different starting conditions. Although it is somewhat less evident due to the different scales of the plots in Figure 2, average path length and average degree converge just as well. Note that the $(*,*,push)$ protocols are unstable and converge very slowly in the growing overlay scenario. We will return to this issue below.

Also note that in the case of the lattice initialization scenario the initial diameter is very large but even in that case we observe rapid convergence to the desirable low diameter topology (Figure 2(b)).

7.2 Randomness

Let us compare the overlays with random graphs in which the view is filled with uniform random samples of the other nodes. The behavior of the protocols we examined shows a rather colorful picture with respect to different graph properties.

In the case of average path length, clustering coefficient and average degree it is clear that protocols $(*,rand,pushpull)$ give us the closest approximation of the random topology, with the tail peer selection being slightly more random (see Figure 2). However, when looking at other aspects, we see a rather different picture. Degree distribution protocols $(rand,head,*)$ are the closest to random distribution while protocols $(*,rand,*)$ are rather far from it (see Figure 3).

In all cases, we can observe that the clustering coefficient is significantly larger than that of the random graph and at the same time the average path length is almost as small. This adds all our overlay topologies to the long list of complex networks observable in nature, biology, sociology, and computer science that have a so-called “small-world” topology [1].

7.3 View selection

The view selection algorithms are significantly different. Head view selection results in a more random degree distribution than the others, and it results in much less autocorrelation of the degree of a fixed node over time (Figures 3 and 4 and Table 2). These properties make the overlays using head view selection much less vulnerable to directed attacks targeting large-degree nodes because there are no nodes with very large degree and the degree of a node changes very quickly anyway. This also means that there are no communication hot-spots in those overlays, which could result in scalability problems.

Also, head view selection repairs the overlay exponentially fast whereas random view selection can at best achieve linear speed, which can hardly be considered scalable (Figure 6). The only scenario when head view selection is not desirable is temporary network partitioning. In that case, with head view selection all partitions will forget about each other very quickly and so quick self-repair becomes a disadvantage. In practical applications the slow and quick self-healing mechanisms should be combined.

7.4 Symmetry of communication

The symmetry of communication is also an important design choice. In particular, push has severe problems dealing with “bottleneck” topologies, like the star-like topology implicitly defined by the growing overlay scenario. In that case, some protocols using the push communication model were not even stable enough with respect to connectivity to participate in the experiments (Table 1), and even those that were included showed very slow convergence. The reason is that nodes that join the network in the growing scenario can get information only if the contact node pushes it to them which is very unlikely to happen because the contact node communicates only once in each cycle, just like the other nodes.

It appears that this parameter plays a more prominent role in characterizing the overall behavior of the various protocols. In general, the performance of push-pull is clearly superior compared to push-only approaches.

7.5 Peer selection

In the case of peer selection we cannot observe drastic differences. In general, applying the tail selection algorithm results in slightly more randomness and slightly slower convergence at the same time. The only scenario in which opting for tail selection results in clear performance degradation is self-healing (Figure 6). In that case, (tail,head,push) converges significantly slower than (rand,head,push), although both converge still very quickly. Also, (tail,rand,push) slowly *increases* the amount of dead links which is especially undesirable.

8 Related work

It would be interesting to conduct a similar experiment for alternative protocols that construct unstructured overlays. A typical example is the one used in the Gnutella peer-to-peer system [9]. In Gnutella, each node gets connected to a number of other nodes, which represent its partial view of the system. In the original version of the protocol there is no explicit mechanism to control communication topology. However, some studies have shown that these networks self-organize into a structure in which the node degree closely follows a power-law distribution [24, 22]. This distribution has a significant impact on the overall system performance as it easily leads to an uneven balance of workload across the nodes that constitute the network.

Another example is the Scamp protocol [8], in which—unlike in the case of the protocols presented in this paper—an explicit attempt is made towards the construction of a random graph topology. Randomness has been evaluated in the context of information dissemination, and it appears that reliability properties come close to what one would see in random graphs.

Several approaches to building more *structured* overlay networks [23, 21, 25] have emerged. These networks typically provide the abstraction of a distributed hash table (DHT) and rely on a global naming scheme allowing each node to maintain tables for highly efficient message routing. The resulting graphs of connections are thus structured by construction. Recent studies analyzed several DHT structures based on graph properties and different failure scenarios [10, 17].

It is worth noting that the assumption of uniform randomness has only fairly recently become subject to discussion when considering large complex networks such as the hyperlinked structure of the Web, or the complex topology of the Internet. Like social and biological networks, the structures of the WWW and the Internet both follow the quite unbalanced power-law degree distribution, which deviates strongly from that of traditional random graphs. These new insights pose several interesting theoretical and practical problems [3].

9 Concluding remarks

If there is any conclusion to draw from our experiments, it is that the gossip-based constructions of overlays through partial views leads to many different topologies, none of which actually resembles traditional random graphs. Instead all these constructions belong to the family of small-world graphs characterized by small diameter and large clustering. When considering the stable properties of various protocols, that is, which emerge from convergent behavior, it also becomes clear that different parameter settings lead to very different properties, which can be exploited according to the needs of the targeted application. For example, a strong self-healing topology may not be appropriate in the presence of temporary network partitions. In many cases, combining different settings will be necessary. Such a combination can, for instance, be achieved by introducing a second view for gossiping membership information. By-and-large, our experiments illustrate that there is still much to do before we reach a point in which we fully understand gossip-based unstructured overlays.

References

- [1] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, Jan. 2002.
- [2] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature*, 406:378–382, 2000.
- [3] A.-L. Barabási. *Linked: the new science of networks*. Perseus, Cambridge, Mass., 2002.
- [4] K. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budi, and Y. Minsky. Bimodal multicast. *ACM Transactions on Computer Systems*, 17(2):41–88, May 1999.
- [5] A. Demers, D. Greene, C. Hauser, W. Irish, and J. Larson. Epidemic algorithms for replicated database maintenance. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 1–12, Vancouver, British Columbia, Canada, Aug. 1987.
- [6] P. T. Eugster, R. Guerraoui, S. B. Handurukande, A.-M. Kermarrec, and P. Kouznetsov. Lightweight probabilistic broadcast. *ACM Transactions on Computer Systems*, 21(4):341–374, 2003.
- [7] P. T. Eugster, R. Guerraoui, A.-M. Kermarrec, and L. Massoulié. From epidemics to distributed computing. *IEEE Computer*. to appear.
- [8] A. Ganesh, A.-M. Kermarrec, and L. Massouli. Peer-to-peer membership management for gossip-based protocols. *IEEE Transactions on Computers*, 52(2), February 2003.
- [9] The Gnutella protocol specification, 2000. <http://dss.clip2.com/GnutellaProtocol04.pdf>.
- [10] K. P. Gummadi, R. Gummadi, S. D. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica. The impact of DHT routing geometry on resilience and proximity. In *Proceedings of ACM SIGCOMM 2003*, pages 381–394, 2003.
- [11] I. Gupta, K. Birman, and R. van Renesse. Fighting fire with fire: using randomized gossip to combat stochastic scalability limits. *Quality and Reliability Engineering International*, 18:165–184, March 2002.
- [12] M. Jelasity, W. Kowalczyk, and M. van Steen. Newscast computing. submitted for publication.
- [13] M. Jelasity, W. Kowalczyk, and M. van Steen. An approach to massively distributed aggregate computing on peer-to-peer networks, 2004. accepted for publication in the proceedings of The 12th Euromicro Conference on Parallel, Distributed and Network based Processing (PDP 2004).
- [14] M. Jelasity and A. Montresor. Epidemic-style proactive aggregation in large overlay networks, 2004. accepted for publication in the proceedings of The 24th International Conference on Distributed Computing Systems (ICDCS 2004).
- [15] M. Jelasity, A. Montresor, and O. Babaoglu. A modular paradigm for building self-organizing peer-to-peer applications. accepted for publication in the proceedings of The 1st International Workshop on Engineering Self-Organising Applications (ESOA 2003).
- [16] A.-M. Kermarrec, L. Massoulié, and A. Ganesh. Probabilistic reliable dissemination in large-scale systems. *IEEE Transactions on Parallel and Distributed Systems*, 14(3), March 2003.
- [17] D. Loguinov, A. Kumar, V. Rai, and S. Ganesh. Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience. In *Proceedings of ACM SIGCOMM 2003*, pages 395–406, 2003.
- [18] M. Newman. Random Graphs as Models of Networks. In S. Bornholdt and H. G. Schuster, editors, *Handbook of Graphs and Networks: From the Genome to the Internet*, chapter 2. John Wiley, New York, NY, 2002.
- [19] R. Pastor-Satorras and A. Vespignani. Epidemic dynamics and endemic states in complex networks. *Physical Review E*, 63:066117, 2001.
- [20] B. Pittel. On spreading a rumour. *SIAM J. Appl. Math.*, 47:213–223, 1987.
- [21] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content-Addressable Network. In *Proceedings of ACM SIGCOMM*, Aug. 2001.
- [22] M. Ripeanu and I. Foster. Mapping the gnutella network: Macroscopic properties of large-scale peer-to-peer systems. In *IPTPS 02*, 2002.
- [23] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *International Conference on Distributed Systems Platforms (Middleware)*, Nov. 2001.
- [24] S. Saroiu, K. P. Gummadi, R. Dunn, S. D. Gribble, and H. M. Levy. An analysis of Internet content delivery systems. In *OSDI'02*, Dec. 2002.

- [25] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In *Proc. ACM SIGCOMM'01*, San Diego, CA, Aug. 2001.
- [26] S. Voulgaris and M. van Steen. An epidemic protocol for managing routing tables in very large peer-to-peer networks. In *Proceedings of the 14th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, (DSOM 2003)*, number 2867 in Lecture Notes in Computer Science. Springer, 2003.
- [27] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.