Network Awareness and Failure Resilience in Self-Organising Overlay Networks

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

Abstract

The growth of peer-to-peer applications on the Internet motivates interest in general purpose overlay networks. The construction of overlays connecting a large population of transient nodes poses several challenges. First, connections in the overlay should reflect the underlying network topology, in order to avoid overloading the network and to allow good application performance. Second, connectivity among active nodes of the overlay should be maintained, even in the presence of high failure rates or when a large proportion of nodes is not active. Finally, the cost of using the overlay should be spread evenly among peer nodes for fairness reasons as well as for the sake of application performance. To preserve scalability, we seek solutions to these issues that can be implemented in a fully decentralized manner and rely only on local knowledge from each node.

In this paper, we propose an algorithm called the localiser which addresses these three key challenges. The localiser refines the overlay in a way that reflects geographic locality so as to reduce network load. Simultaneously, it helps to evenly balance the number of neighbours of each node in the overlay, thereby sharing the load evenly as well as improving the resilience to random node failures or disconnections. The proposed algorithm is presented and evaluated in the context of an unstructured peer-to-peer overlay network produced using the Scamp protocol. We provide a theoretical analysis of the various aspects of the algorithm. Simulation results based on a realistic network topology model confirm the analysis and demonstrate the localiser efficiency.

Keywords unstructured overlay networks, peer-to-peer, random graphs, resilience to failure, network locality, load balancing.

1 Introduction and overview

Peer-to-peer applications currently constitute one of the fastest growing uses of the Internet. A challenging research problem is to develop overlay architectures that can support such applications without overloading network resources. Features of peer-to-peer overlay networks such as the transience of users and the absence of powerful central servers motivate fully decentralized schemes that do not require global knowledge of network topology or membership, and that are robust to node failures or disconnections. Scalability is crucial in this context, and overlays must be built ensuring that the potential load on each node grows slowly as the overlay size increases while the global load is evenly balanced between nodes. Other objectives include low latency, fault tolerance, and ease of reconfiguration when failures occur.

Several recent approaches to building structured overlay networks [16, 15, 17] employ the abstraction of a distributed hash table (DHT). DHT schemes typically combine a global naming scheme based on hashing with the maintenance of hierarchically structured routing tables using the assigned names. This rich hierarchical structure of the name space can be leveraged in several ways by applications. Since node names typically don't reflect location in the network topology, additional mechanisms are needed to ensure geographically efficient routing in these systems [16, 20].

The work in this paper differs in considering unstructured overlays. Such overlays do not rely on global naming or any additional hierarchical structure and are currently used in Internet-wide deployed systems such as Gnutella [10]. Several schemes have been proposed for building such overlays [8, 9, 11]. The overlays may be used to propagate information either by flooding, or using epidemic protocols [18, 8] or via application-level multicast [11, 3]. In this paper, we consider the Scamp protocol [9], in which the only information maintained at each node is the list of addresses of a subset of other nodes called its neighbours. Scamp provides probabilistic guarantees on global connectivity in the presence of failures by ensuring that each node has a sufficiently large degree, where degree refers to the number of its neighbours in the overlay. The node degree required to ensure good probabilistic guarantees on connectivity turns out to be logarithmic in the overlay size [12]. In SCAMP, as nodes join and leave the system, the overlay adapts automatically so that the mean degree of a node is $c\log(N)$, where N is the number of nodes in the system, and c is a design parameter whose choice determines the extent of resilience to node failures. Since each node only knows about its neighbours in the overlay network, the memory requirements on a node grow only logarithmically in the overlay size. Note that the adaptation of the degrees to $c\log(N)$ takes place without the nodes having to explicitly know the system size, N.

The Scamp protocol is oblivious to the underlying network topology. Hence, communication between peers on the Scamp overlay tends to impose a high load on the network, thus limiting its applicability in wide-area settings. Moreover, the mean node degree converges to $\log(N)$ but the degrees of individual nodes are not tightly constrained, so some nodes may be responsible for communicating with many more peers than others. If these nodes sit behind low-bandwidth connections, application performance may be compromised.

In this paper, we propose a mechanism to optimize unstructured overlays according to a proximity criterion in order to reduce network load. The same mechanism can be used, at virtually no extra computation or communication cost, to balance node degrees. Balancing degrees not only helps to balance the load, but more importantly improves the resilience of the system to link and node failures. (If all nodes are not equal in their capabilities, either in terms of the load they can handle or in the periods for which they are disconnected, then we may wish to have unbalanced degrees in a pre-specified fashion. Our mechanism can be adapted to this end, with no increase in complexity.) All these features are integrated into a single algorithm called the *localiser*, which functions in a fully decentralized fashion and requires no global knowledge. We evaluate this algorithm in the context of the Scamp protocol although it can be applied more generally to peer-to-peer overlays.

The rest of the paper is organized as follows. In Section 2, we set out our design requirements for overlay networks and give an overview of the Scamp protocol. Section 3 presents the localiser algorithm and its analysis. We discuss the resilience to failures in Section 4. We present performance results in Section 5. Section 6 contains a discussion of related work and conclusions.

2 Unstructured peer-to-peer overlay networks

2.1 Design requirements

The basic property required of an overlay network is that it be connected, so that any two peers can communicate over it. Moreover, connectivity should be maintained in the presence of failures or temporary disconnections of some fraction of member nodes, and the communication latency between any two members should not be too large. While a fully connected overlay would meet these requirements, it requires global knowledge of membership which is obviously impractical in large systems.

We shall model the overlay as a graph with nodes i representing members, and a directed edge (i,j) representing that i knows j. If the “who knows who” relation is symmetric, the edges are taken to be undirected. In the rest of the paper, we call the nodes to which a node is connected in the overlay its neighbours. The degree of a node is the number of neighbours it has. If the graph is directed, the in-degree of node i is the number of nodes that know i, while its out-degree is the number of nodes that i knows.

A classical random graph model, which was studied in detail by Erdös and Renyi, has a collection of N nodes; an edge is present between every pair of nodes with probability p (which may depend on N), independent of other edges. If the graph is undirected, the mean degree of a node is $(N-1)p$, whereas if it is directed, $(N-1)p$ is the mean in-degree or out-degree. In the undirected case, Erdös and Renyi [7] showed that there is a threshold for connectivity at a mean degree of $\log(N)$: precisely, if the mean degree is $\log N + x$, then the probability of connectivity goes to $e^{-e^{-x}}$ as $N \to \infty$.[1] It was shown by Bollobàs [5, Theorem 10.17] that the diameter of such a random graph is of order $\log N/\log\log N$.

These analytical results suggest that the membership protocol should be designed so as to provide each member with a set of neighbours whose size is of order $\log N$. In that case, the memory requirements on each member grow slowly in the overlay size, but the overlay network nevertheless has the desired properties.

In addition, it is desirable that the set of neighbours at a node consist mainly of nearby nodes, in an appropriate network proximity metric. This is needed to ensure that communication on the overlay doesn't impose a high network load. Finally, we shall require the degree at all nodes to be of approximately the same size, both in order to balance load and to improve the resilience to failures.

[1]This was extended by Ball and Barbour [1] to random directed graph models with a wide range of degree distributions. They showed a similar threshold at a mean degree of $\log(N)$ for the probability that there is a directed path from any specified node to all other nodes.

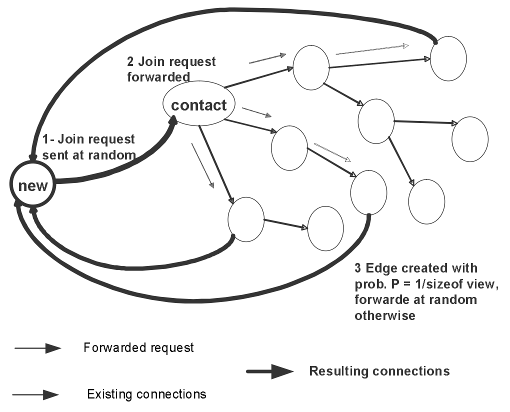
2 Join request forwarded
contact
1- Join request sent at random
new
3 Edge created with prob. P = 1/sizeof view, forwarde at random otherwise
Forwarded request
Resulting connections
Existing connections

Figure 1. Joining algorithm

We begin with an overview of the Scamp protocol, which achieves degrees of the desired size on average. In subsequent sections, we show how to incorporate locality and to balance the degrees at different nodes.

2.2 Scamp

SCAMP is a fully decentralized self-organizing membership protocol for building unstructured overlay networks. It ensures that the mean degree scales as $c\log(N)$, for a given design parameter c, where N is the overlay size. This is done without maintaining any global information about the overlay size, and is made possible by 'bootstrapping' on the current mean degree when adding new members.

The Scamp joining algorithm, illustrated in Figure 1 works as follows

1. New nodes join the system by contacting an arbitrary member of the system. The connection between the new member and the contact is used to connect the new member to the overlay.

2. When a node receives a join request, it forwards the new node ID to all its neighbours in the overlay. It forwards $c-1$ additional copies of the join request to randomly chosen neighbours.

3. When a node receives a forwarded join request, it integrates the new member as one of its neighbours with a probability p which depends on its degree in such a way that nodes with a low degree are more likely to connect to the new member. Otherwise, it forwards the request to a randomly chosen neighbour.

When N nodes have joined the system, the overlay constructed by this protocol is connected, and each node is connected on average to $c\log(N)$ other nodes. A full description of the protocol, including an analysis and performance evaluation, can be found in [9].

3 Network awareness and load balancing

Motivation In Scamp, all connections are treated as if they are equivalent, and so the resulting overlay does not reflect the underlying network topology. Since it would be more efficient if each node were to have local neighbours rather than remote ones, our first objective is to favour low cost connections, where cost refers to a given network performance criterion (such as delay or spare capacity). We seek to do this without compromising the probabilistic guarantees of connectivity. Secondly, Scamp achieves a mean node degree of $c\log(N)$ but does not evenly balance the degrees of nodes. The second objective is to have the degree of each node be of nearly equal size. This should significantly increase the resilience to failures. We propose a Metropolis-type algorithm, called the localiser, to achieve both these goals. Note that if nodes have different inherent capabilities, we may wish to skew the degrees of individual nodes in a pre-determined fashion (while keeping the mean at $c\log(N)$) rather than let this happen at random. The localiser can easily be modified to achieve this, as described below.

Although the localiser algorithm is presented and evaluated in the context of Scamp, it can be applied to any unstructured overlay.

Localiser The localiser relies on a Metropolis scheme [14] (like simulated annealing at fixed temperature). This is a simple iterative scheme for minimising a function f on a domain D. Starting from any point $x \in D$, a move to a nearby point y is proposed, for example by sampling the step $y-x$ from a specified distribution. The move is then accepted with some probability which is decreasing in $f(y)-f(x)$. The reason for allowing moves that may increase f is to avoid becoming trapped at a local minimum. If the move is rejected, the new point is x. The process is repeated from the new point.

We adapt this basic idea to our problem by choosing an energy function incorporating our objectives. The energy function is

$$E(G) = w \sum_{i \in V(G)} d_i^2 + \sum_{(i,j)\in E(G)} c(i,j),$$

where the sum is taken over all vertices i and edges (i,j) in G, w is a weight parameter and $c(i,j)$ is a measure of the cost of communication between i and j. For example, $c(i,j)$ could be the ping time or a more complex measure incorporating bandwidth availability on the route between i and j. Note that this is a global function incorporating the degrees of all nodes and the cost of all links. The essential feature that enables us to obtain a decentralized protocol is that the difference in energy between "neighbouring" con-

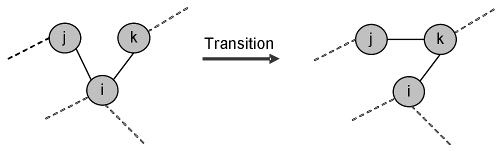

Figure 2. Effect of a local transition

figurations can be computed locally, as shown below. We now give a detailed description of the protocol.

Algorithm Each node maintains the IP address of its neighbours in the overlay in a data structure called `Degree`. We consider symmetric connections here[2]. Periodically, and at unit rate, each node i performs the following steps:

1. Choose two of its neighbours, node j and node k uniformly at random from `Degree`, and measure the link costs $c(i,j)$ and $c(i,k)$;

2. Send messages to node j and node k, which send back respectively $d(j)$ and $d(k)$, their degrees. In addition, node j sends back its estimate of $c(j,k)$;

3. Evaluate locally the cost of replacing link (i,j) with link (j,k) as depicted on Figure 2. The cost is the energy difference between these configurations, which is $\Delta E = 2w(d_k - d_i + 1) + c(j,k) - c(i,j)$.

4. Perform the transition, as depicted on Figure 2, with probability $p = \min\left(\left(e^{-\Delta E/T}\frac{d_i(d_i-1)}{d_k(d_k+1)}\right),1\right)$. To this end, i sends messages to j and k asking them to establish connections with k and j respectively. In addition i and j get disconnected (after receiving an acknowledgement that the connections between j and k has been established).

Note that the algorithm makes changes which would increase the energy with a positive probability, and hence avoids getting stuck at local minima of the energy function. Also note that the number of edges, and hence the mean degree, is conserved since the algorithm only moves edges.

The localiser is parameterized by w and T:

1. The parameter w specifies the emphasis placed on degree balancing relative to locality. If $w = 0$, the graph is optimized only to take into account network proximity. Evaluation has shown that the resulting maximum degree is large compared to the mean degree. As $w \to \infty$, balancing node degrees takes precedence over improving locality.

2. The parameter T (used to compute the probability of performing a transition) is called temperature and specifies a trade-off between accuracy - how close we get to the optimal configuration - and speed of convergence. Small values of T ensure that the system is more likely to concentrate on low energy configurations but can lead to slower convergence.

[2]The edges are oriented in the initial Scamp protocol.

Remarks It is easy to modify the algorithm to account for different inherent capabilities of the nodes. Each node i chooses a weight η_i within some pre-specified range to reflect its capability. The energy function is specified as

$$E(G) = w\sum_{i\in V(G)}(d_i/\eta_i)^2 + \sum_{(i,j)\in E(G)} c(i,j).$$

We now use the same algorithm as above, with the obvious modification that

$$\Delta E = 2w\left(\frac{d_k}{\eta_k} - \frac{d_i}{\eta_i} + \frac{1}{2\eta_k^2} + \frac{1}{2\eta_i^2}\right) + c(j,k) - c(i,j).$$

The effect of the modification is that the algorithm seeks to equalize d_i/η_i instead of d_i.

For the sake of clarity, we have presented the localiser algorithm as acting on a overlay that has previously been constructed using Scamp or some other protocol. In practice, it would obviously be more efficient to integrate the localiser into the overlay construction protocol.

Analysis The localiser algorithm is a particular instance of the Metropolis algorithm [14] (see also the survey in [6]). In particular, it defines a reversible Markov chain on the space of connected graphs on N nodes with M arcs. Denote by $G(N,M)$ the space of labelled graphs on N nodes with M arcs. The stationary distribution of this Markov chain is given by the Gibbs distribution on $G(N,M)$ defined by

$$\pi(G) = \frac{1}{Z_T}e^{-E(G)/T}\mathbf{1}_{G \text{ is connected}}, \qquad (1)$$

where Z_T is a normalization constant.

In other words, if the algorithm is stopped after a sufficiently long time, the resulting overlay is a random graph G whose probability distribution is close to that given in (1). If we take $T = \infty$, then $\pi(G)$ is the uniform distribution on all connected graphs with N nodes and M edges. This is the same as the distribution in the classical Erdös-Renyi model, conditioned on having M edges and on being connected; it takes no account of locality or of degree balance. As T decreases, the distribution $\pi(G)$ increasingly favours low energy configurations, namely those with good locality and with balanced degrees.

We conclude this section with an illustrative example of 100 points placed uniformly at random on the unit square

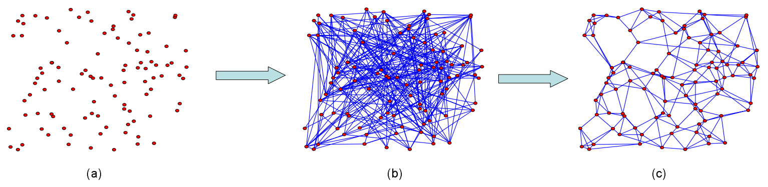

Figure 3. LOCALISER**: (a) Initial system; (b) Overlay produced by Scamp; (c) Overlay refined by the localiser**

(Figure 3(a)). Figure 3(b) depicts an unstructured overlay produced using the standard Scamp protocol. The localiser refines the basic overlay by favouring local nodes and homogenizing the degree at the same time (Figure 3(c)). Note that the number of edges in Figures 3(b) and 3(c) is the same, and their apparent sparsity in the latter is entirely due to the edges being much shorter on average.

4 Resilience to failures

Nodes might leave and join the system dynamically. The overlay leave protocol must maintain the full connectivity of the overlay as well as maintaining the degree scaling. When a node leaves the system voluntarily in the Scamp protocol, it sends a leave message and the algorithm ensures that the connectivity is maintained [9]. The protocol remains local and only the leaving node and its direct neighbours in the graph are involved in the leave process. In addition a lease mechanism is used to handle node failures. Each node joining has a finite lifetime called its lease. This could be set either by individual nodes at the time they join the overlay, or could be a property of the overlay which is imposed on all members. When a node's lease expires, every neighbour disconnects it. It is the responsibility of each node to re-join at the time that its lease expires. Non-faulty nodes do not let their lease expire and so connectivity is not compromised.

The connectivity analysis of random graphs can easily be extended to account for link and node failures [12]. For example, if links fail with probability ε independently of each other, then the graph remains connected after failures if the initial mean degree is taken to be $(\log(N))/(1-\varepsilon)$. Such results have been used in probabilistic gossip-based algorithms [4].

The overlay constructed by the Scamp protocol can tolerate failure rates up to $(c-1)/c$ before losing connectivity. Moreover, as the failure rate increases, connectivity is initially lost because of the isolation of a small number of nodes which, due to the random nature of the protocol, started out with a smaller degree than average. By balancing the node degrees, we can avoid this phenomenon and thereby tolerate a higher failure rate before losing connectivity.

For the result below, we ignore the geographic criterion by setting $c(\cdot,\cdot) \equiv 0$ and focus on the effect of balancing the degrees.

Theorem 1 *Let G be a random graph with distribution given by (1), for any fixed $T > 0$, with $M = c \log N$ and c a given constant. Let ε be the probability of link failure, and assume that links fail independently of each other. Let $\delta > 0$ be fixed. If $c > (1+\delta)/(-\log(\varepsilon))$, then the probability that the graph becomes disconnected due to link failures goes to zero as $N \to \infty$. If $c < (1-\delta)/(-\log(\varepsilon))$, then the probability of disconnection goes to one as $N \to \infty$.*

For space reasons, we do not provide the proof in this paper.

5 Performance evaluation

In this section, we present some simulation results demonstrating the effectiveness of the localiser algorithm on an unstructured overlay.

5.1 Experimental setting

We use a simple packet-level discrete event simulator and run the simulations on realistic network topologies. We use two network topology models. Each topology is composed of a set of routers to which nodes are randomly attached. GT-5050 is a topology generated randomly according to the Georgia Tech [19] transit-stub model, composed of 5050 core nodes. A LAN attached is attached to each core node. Each LAN has a star topology and is composed of 100 nodes on average. The second topology, called *Corp-Net* relies on 298 routers and is generated using real measurements of the world-wide Microsoft Corporate Network. We evaluate a 50,000 node system in the two topologies Link delays are modeled by simply assigning a propagation delay of 1ms to each LAN link and 40.5ms to each core link. The distances used in the localiser between nodes are computed using these delays in the simulator. The temperature is set to 1 in all experiments. The results presented

(W, T, NbIt)	(0,0,0)	(10,1,100)	(10,1,1000)	(10,1,5000)	(50,1,100)	(50,1,1000)
Max degree	63	23	21	20	20	20

Figure 4. Max value of the out-degree in a 50,000 node system on CorpNet, mean value is approximately 20

(W, T, NbIt)	(0,0,0)	(10,1,100)	(10,1,1000)	(10,1,5000)
Max degree	61.2	38	27.8	27

Figure 5. Max value of the out degree in a 50,000 node system on Top-5050, mean value is approximately 20

are the average of the results obtained in 10 simulations for each configuration.

We evaluate the localiser along three metrics *(i)* to evaluate the impact of the degree balancing mechanism, we measure the mean and maximum degree, varying the weight w and the number of iterations; *(ii)* to assess to what extent the localiser captures the underlying network topology, we measure the mean edge length, namely the mean distance between a node and its neighbours in the overlay; *(iii)* finally, to assess the impact of the localiser on the resilience to failures, we measure the number of disconnected nodes as the number of faulty nodes increases. We compare the performance of the basic overlay produced by Scamp (using $c = 2$) with that of the refined overlay obtained by using Scamp and then applying the localiser algorithm.

5.2 Degree Balancing mechanism

To evaluate the degree balancing mechanism, we ran experiments varying the value of w and the number of iterations of the localiser algorithm per node. In the rest of this section, the parameter values are given in the format (w, T, number of iterations per node), with (0,0,0) referring to a standard overlay built using Scamp. Figure 4 shows the maximum degree (maximum number of neighbours) in a 50,000 node system on *CorpNet*. The mean value of the degree is $2 * \log(50000) \approx 22$, and is not affected by the localiser. We tried two different values for w. When w is set to 50, the localiser acts almost exclusively as a degree balancing mechanism. Results show that the localiser is indeed very effective in homogenizing node degrees quickly, and the maximum degree gets very close to the mean value as the number of iterations increases. However, this value for w turns out to show poor performance on the network locality metric. In contrast, the choice $w = 10$ does only slightly worse on degree balance, but does much better in terms of network locality. Figure 5 presents the results in GT-5050, which are very similar.

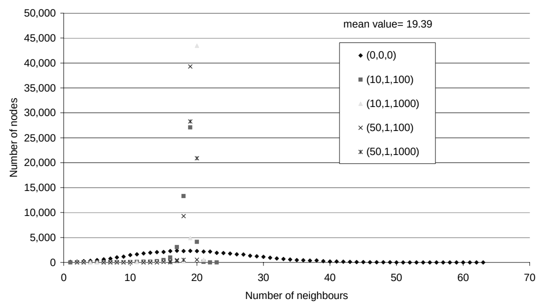

Figure 6. Degree distribution in a 50,000 node overlay on Corp Net (1 run representative)

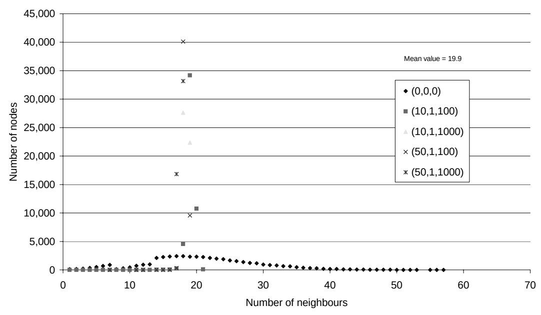

Figure 7. Degree distribution in a 50,000 node overlay on Top-5050 (1 run representative)

(W, T, NbIt)	(0,0,0)	(10,1,100)	(10,1,1000)	(10,1,5000)	(50,1,100)	(50,1,1000)	(50, 1, 5000)
Mean distance	266	225	141	66	253	224	215
Standard deviation	250	208	124	51	237	207	199

Figure 8. Mean and standard deviation of distances to neighbours in a 50,000 nodes system on Top-600

(W, T, NbIt)	(0,0,0)	(10,1,100)	(10,1,1000)	(10,1,5000)
Mean distance	483	408	290	196
Stdv	472	396	274	182

Figure 9. Mean and standard deviation of distances to neighbours in a 50,000 nodes system on Top-5050

The results show that the maximum degree gets close to the mean value after a reasonable number of iterations and so the localiser is indeed effective in balancing the load. It should be noted that if the localiser is integrated into the overlay building protocol, the number of iterations needed to reach a good configuration should be significantly reduced.

Figures 6 and 7 depict the distribution of the degree for various configurations. These figures demonstrate that the distribution becomes sharply concentrated around the mean value when the localiser is used, thereby decreasing the sensitivity to failures. For instance, in the (10,1,1000), (50,100) and (50,1,1000) configurations in both topologies, the number of neighbours takes only one of two values, namely the two integers between which the mean degree lies. In other words, 1000 iterations suffice in these examples to achieve perfect degree balancing, and 100 iterations suffice when w is large, i.e., the emphasis is on balancing degrees. In practice, we don't require perfect degree balancing - adequate balancing is achieved after 100 iterations even for $w = 10$, where the emphasis is more on improving locality.

5.3 Network awareness

The second objective of the localiser is to enable the overlay to reflect the network topology. To evaluate its performance on this goal, we measure the distance from each node to its neighbours in the overlay. Figures 8 and 9 present the mean distance and the standard deviation in a 50,000 node system on *CorpNet* and GT-5050 respectively. We see from Figure 8 that, when w is set to 50, the localiser mainly acts as a degree balancing mechanism and the network locality is captured less well than when w is set to 10. We see that both the mean value and the standard deviation decrease as the number of iterations per node increases. The results demonstrate that the localiser is effective in capturing the underlying network topology. The fact that distances remain much bigger than the average distance between two nodes in a LAN is explained by the fact that LANs are sparsely populated. To verify this, we conducted an experiment on a fully populated network. The network was generated using a Georgia Tech topology composed of 600 core nodes, to each of which a 100-node LAN is attached. We ran an experiment where all 60,000 nodes belong to the group on which the overlay is constructed. The mean distance to neighbours in the overlay converges towards the distance between two node in a LAN (2); it was 15 after 10000 iterations and 5 after 20000 iterations.

5.4 Resilience to failures

In order to assess the resilience to failures, we measure the number of nodes that become disconnected as the number of faulty nodes increases from 0 to 25,000 (50% of the nodes) in steps of 250. In order to evaluate the number of disconnected nodes, we use a flooding message initiated at a random live node and compute the number of live nodes receiving the flooded message.

In figures 10 and 11, we plot the number of disconnected nodes against the number of faulty nodes in CorpNet and GT-5050 respectively. Note that it is possible that a live node disconnected at iteration i is not counted as a disconnected node at iteration $i+1$ because it has become faulty. This explains why the number of disconnected nodes in Figures 10 and 11 can decrease when the number of faulty nodes increases. Since the source of the flooding message is chosen at random among live nodes, the source node might be isolated and disconnected from the main cluster. In that case, the message dies out and no node receives the message. For the sake of clarity of the figures, we exclude simulation runs where this happened. However, it should be noted that such cases were rare, and the localiser did not af-

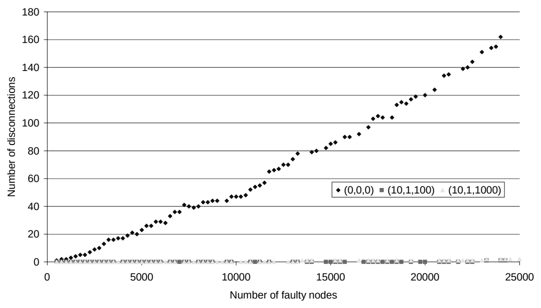

Figure 10. Resilience to failures in a 50,000 node overlay on Top-Corp

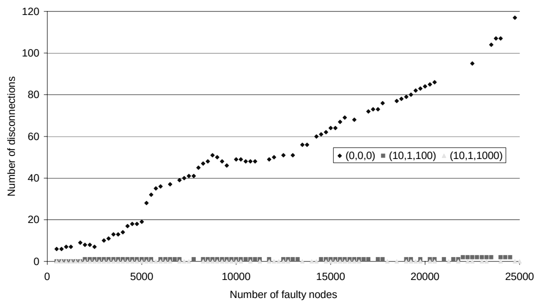

Figure 11. Resilience to failures in a 50,000 node overlay on Top-5050

fect the proportion of experiments in which this happened.

The first point to note from the figures is the good connectivity of the basic overlay produced by Scamp; we have chosen $c = 2$, so the Scamp overlay should be able to tolerate up to 50% failures. Indeed, the fraction of live nodes disconnected stays below 0.5% when up to 50% of the nodes fail. Secondly, both figures demonstrate that the localiser algorithm improves the resilience to failures significantly, after just a small number of iterations. The impact of the localiser increases with the number of faulty nodes. For example, in GT-5050 when 50% of the nodes are faulty, 0.40% and 0.07% of live nodes are disconnected in, respectively, the standard overlay and the localiser (10,1,100). There were no disconnections observed in the localiser (10,1,1000) and(10,1,5000) configurations. For this reason, we have not shown results for the (10,1,5000) configuration. Disconnections (at a rate of 0.01%) were observed only when more than 60% of nodes are faulty.

6 Related work and conclusion

We begin with a brief review of different approaches for taking locality into account when constructing overlay networks. Some structured overlay networks such as CAN [15], Pastry [16] and Tapestry[20] integrate network proximity into the maintenance of the overlay. In an extension to the initial CAN protocol, nodes measure their distance to a set of landmark nodes. They use these distances to compute their relative position in the overlay, thus partially reflecting the Internet topology. In Tapestry and Pastry, nodes choose local nodes as much as possible to populate their routing tables. Our work differs in that it applies to unstructured networks, where the absence of structure and hierarchy provides greater freedom in updating connections.

The NICE infrastructure [2, 3] relies on a hierarchical structure which is updated to reflect the network topology as nodes join. The root of the hierarchy is in charge of dealing with every single join request and propagating it down the hierarchy in such a way that new nodes join the right low level cluster to reflect the network topology. Such a scheme limits the scalability of the approach. In Narada, [11], an unstructured optimized overlay is built in a decentralized way. This mesh is then optimized according to several network metrics (load, distance, end node capacity) by computing a utility function that is continually updated. Narada requires each node to maintain the global membership information and to update it as nodes leaves and join the system. Such a scheme is feasible in the small to mid-size systems targeted by Narada, but would not scale to very large system sizes.

Directional Gossip [13] creates an overlay targeted at wide-area networks. It takes account of the network topology by computing a weight for each neighbour reflecting the connectivity and the network topology. However, this overlay construction leads to a static hierarchy sensitive to failures.

In this paper, we have proposed a localiser algorithm which optimizes unstructured large-scale overlay networks. The proposed algorithm serves multiple purposes: *(i)* it reflects the network topology by reshaping the overlay via a Metropolis scheme; *(ii)* it achieves load balancing by sharing the responsibility for connections evenly among the system nodes and finally *(iii)* it significantly improves the resilience to failures. The localiser is fully decentralized and only relies on local information available at each node.

We present an analysis of the localiser algorithm, as well as detailed simulation results which demonstrate the effectiveness of the localiser along the three metrics that we targeted (network proximity, load balancing and resilience to failures). Future work includes the study of the localiser in the presence of high churn rate. We are currently working on efficient application-level multicast based on unstructured overlays optimized using the localiser.

References

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

[2] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In *Proc. ACM SIGCOMM*, Pittsburgh, PA, USA, August 2002.

[3] S. Banerjee, S. Lee, B. Bhattacharjee, and A. Srinivasan. Resilient multicast using overlays. In *Proc. ACM SIGMETRICS*, San Diego, CA, USA, June 2003.

[4] 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.

[5] B. Bollobs. *Random Graphs*. Cambridge University Press, 2001.

[6] P. Brémaud. *Markov chains*. Springer, 1999.

[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. *ACM Transactions on Computer Systems*, To appear, 2003.

[9] A. Ganesh, A.-M. Kermarrec, and L. Massoulié. Peer-to-peer membership management for gossip-based protocols. *IEEE Transactions on Computers*, 52(2), February 2003.

[10] The Gnutella protocol specification, 2000. http://dss.clip2.com/GnutellaProtocol04.pdf.

[11] Yang hua Chu, Sanjay G. Rao, and Hui Zhang. A case for end system multicast. *IEEE Journal on Selected Areas in Communications (JSAC)*, 20(8), October 2002.

[12] A.-M. Kermarrec, L. Massoulié, and A.J. Ganesh. Probabilistic reliable dissemination in large-scale systems. *IEEE Transactions on Parallel and Distributed Systems*, 14(3), March 2003.

[13] 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.

[14] N. Metropolis, M. N. Rosenbluth, A. W. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[15] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content-Addressable Network. In *Proc. ACM SIGCOMM*, San Diego, CA, USA, August 2001.

[16] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. International Conference on Distributed Systems Platforms (Middleware)*, Heidelberg, Germany, November 2001.

[17] 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*, San Diego, CA, USA, August 2001.

[18] R. van Renesse, K. P. Birman, and W. Vogels. Astrolabe: A robust and scalable technology for distributed systems monitoring, management, and data mining. *ACM Transactions on Computer Systems*, 21(3), 2003.

[19] E. Zegura, K. Calvert, and S. Bhattacharjee. How to model an internetwork. In *Proc. IEEE INFOCOM*, San Francisco, CA, USA, 1996.

[20] B.Y. Zhao, J.D. Kubiatowicz, and A.D. Joseph. Tapestry: An infrastructure for fault-resilient wide-area location and routing. Technical Report UCB//CSD-01-1141, U. C. Berkeley, April 2001.