

TopDNS: Experiences Building Internet-Based Topologies with GNP

Gert Pfeifer and Christof Fetzer
TU Dresden
Faculty of Computer Science
01062 Dresden, Germany
{firstname.lastname}@inf.tu-dresden.de

Abstract

Building realistic Internet topologies is an important step towards predicting the behavior of new distributed applications and protocols. We are creating topologies that predict the distances between identifiable nodes in the Internet as it is today. These nodes are not anonymous average Internet nodes, but are known by name and network address. We are using a landmark and network coordinate based approach and evaluate the characteristics of the topologies found. We provide hints, how to select important parameters for such topology generators.

1. Introduction

Designing overlay solutions which can compete with IP-based routing is difficult [2, 5]. Our motivation is to show that our overlay network DNSPastry [10] can exploit the DNS name structure of hostnames to connect nodes in an overlay in a way that improves the efficiency of overlay routing. So far we have been using transit-stub-topologies [16] and random ping benchmarks. However, we would like to evaluate if our overlay can provide a DNS service to increase the dependability based on the fault tolerance and flexibility of peer-to-peer networks as proposed in [2] and [9].

As a workload for such benchmarks we use traces of DNS requests recorded at the servers of our department. The main problem is that the realistic characteristics of the traces, i.e., which host name is used how frequently, must be mapped to the abstract instances of nodes and networks in the topology that we are using for the simulation. The easiest way to do that is to have real node names in the topology. A request in the DNS trace would then be routed to a node representing this name. This might be the DNS server for this zone, or even the node itself, if the name is used for routing instead of resolving it to an IP in the first place.

To evaluate the performance of DNSPastry, we used the

stretch, i.e., overlay routing delay over IP routing delay [5, 10], as a representation of the penalty that comes with overlay routing. Prerequisite for the computation of stretch is knowledge about the IP-routing delay between any pair of nodes in the topology. Hence, we do not require our topology to represent the actual route a packet takes between two nodes. Instead, the distance between real identifiable hosts in the Internet is sufficient. A host is identifiable, iff it is known by name and IP address.

TopDNS is a topology generator that fulfills these requirements. In this paper, we describe the experiences we made while building and using it. It is able to provide very precise topologies, where precision is defined as similarity to the Internet at the point in time when the landmarks made their measurements.

The rest of the paper is organized as follows: Section 2 discusses related work. Our system architecture is presented in Section 3. In Section 4 we discuss our results and experiences and Section 5 concludes.

2. Related Work

Many topologies are created using characteristics of the Internet that are statistically analyzed and used to represent its intrinsic properties. Examples of such characteristics are the AS structure and routes represented in the BGP routing tables, in-degree and out-degree of routers, churn in routing tables, etc. A very common way to represent the AS based hierarchical structure of the Internet are transit-stub-topologies [1, 16]. We have used such topologies in our previous performance analysis of DNSPastry [10]. The main problem with such topologies is, that it is nearly impossible to run benchmarks which use real Internet addresses on them, since there is no way to map existing Internet nodes to nodes in such a topology. A workload for such topologies would also have to be abstract and represent the statistical properties of a typical workload.

Vivaldi [3] is a very efficient distributed network coordinate system. Each participating node can compute its coor-

ordinates using just a small number of latency measurements. Each node in Vivaldi computes its coordinates by simulating its position in a network of physical springs. Such a network behaves in a way that all springs are stretched to a point that reduces the overall energy in the network. Therefore, springs with a very high energy, i.e. a high distance between neighbors in an overlay, might exist contributing to the overall low energy of springs, i.e. low distance between neighbors. This system is used, e.g., in Bittorrent clients. The disadvantage of such an algorithm is, that each node must participate to do measurements, which is in our case not possible. Vivaldi is usable with different metrics. The authors of [3] reached best results using 2d coordinates together with a height.

Several schemes to estimate Internet latencies use landmarks. Landmarks are often nodes to which a remote login is possible, hence they actively participate in the distance computations by contributing measurements of round trip times.

Internet Distance Maps (IDMaps) [4] places landmark nodes at well distributed locations in the Internet. It estimates the latency between two nodes A and B as the latency from A to its closest landmark added to the latency from B to its closest landmark plus the latency between the two landmarks. With an increasing number of landmarks, the error of distance estimation can be reduced. One problem with IDMaps is that a client node has to measure distances to all landmark to identify the closest among them and a rather high number is needed to achieve good precision.

Dynamic Distance Maps (DDM) [14] is comparable to IDMaps but it is utilizing a more sophisticated way to find appropriate landmarks. DDM organizes them hierarchically, and a client node traverses the hierarchy top-down to locate a near-by candidate.

M-coop [13] utilizes a network of nodes linked in a way that mimics the autonomous system (AS) graph extracted from BGP routing information. In contrast to IDMaps, each node measures distances only to a small number of other nodes. When an distance between two nodes is to be estimated, a path containing several measurements is created to provide it. The performance and quality is comparable to IDMaps.

Global Network Positioning (GNP) [7] represents the topology as a Euclidian space. Each node is positioned using a set of coordinates. The authors measured that using a 7-dimensional Euclidean space, in 90% of the cases, GNP can predict the Internet distances among a globally distributed set of hosts with less than 50% error. GNP also uses a set of landmarks. The number of available landmarks limits the number of dimensions in the Euclidian space. Measurements of [7] show that GNP reaches a better precision than IDMaps.

King [6] is similar to IDMaps and M-coop, but uses DNS

servers as landmarks. For each pair of nodes that is measured, the distance between the nameservers that are authoritative for their names is measured. Say we want to estimate the distance between node A and B from our own node C. Therefore, we measure the distance from C to A's name server and send a request for a non-existing name to this server. The name must be chosen in a way that it does not produce a cache hit and that it is served by B's name server. The answer time is measured. It contains a round trip between C and A's name server, which we can measure directly and a round trip between C and B's name server. Hence, we can calculate the distance between A's and B's name servers. The assumption is that this distance is very similar to the distance between A and B. However, the mushrooming of content distribution networks and the world-wide replication of name servers reduces the probability that this assumption holds. The second problem is that we cannot expect name servers to be recursion enabled to everyone as we experienced in your studies for [11].

3. System Design

TopDNS is using DNS traces to collect node names. To assign coordinates to these nodes, it uses GNP. We are using Planet-Lab [12] nodes as landmarks. A python script is exercised on each of them which first determines the distances to all other landmarks and then measures the distance to each of the hosts found in the DNS trace. To collect the data we use a *MySQL* database. It provides the names of the landmarks, as well as all names found in the DNS trace. All landmark nodes ping all nodes in the trace and store the distances, i.e., round trip times, in the data base. The complexity of the topology generation is $O(N)$, for N unique nodes in the traces. A significant part of the work, i.e. measuring the round trip times, is done in parallel.

Using DNS traces allows us to find hosts that are actually in use. This gives us the advantage that our topology has a strong emphasis on machines that are important for Internet users.

To select landmarks we used CoMon [8] to find alive lightly loaded Planet-Lab nodes. However, it turned out that some of them nevertheless had more or less permanent problems. The overall poor dependability of Planet-Lab nodes as observed by Warns et al. in [15] is a general problem that limits the number of available landmarks significantly. Some of the problems we encountered are:

- a firewalls blocked *MySQL* traffic
- a gateway blocked traffic between the commercial Internet and Internet2
- *YUM* problems caused *iptables* not to be started (Planet-Lab nodes are running Fedora Linux)

- slow or overloaded nodes produced measurements with a high deviation
- nodes are not up and running during the complete measurement process

To improve the quality of our topologies, we took several measures to get rid of poor measurements:

- Landmarks with too few measurements were excluded.
- Landmarks with a too high standard deviation in the round trip times to nodes in our own network were excluded.
- We used a genetic algorithm to find a good selection of landmark nodes. Starting with a random selection we mutated 1/8 of the selection of landmarks a dozen times. Then we evaluated the fitness of the generation. At the end we used the best generation found so far. Of course, this algorithm does not guarantee an optimal solution, but it is easy to run in parallel and it can be seen, that after a rather small number of generations (5-10) hardly any further improvement can be reached.

From the remaining landmarks, we took the measurements and used GNP to calculate node coordinates. We tried to find the best number of dimensions and landmarks to get a precise, i.e., realistic, topology (Section 4). However, this optimization is limited by the number of landmarks available.

4. Results

When running TopDNS, there are two things to optimize: (1) to maximize the number of nodes in the topology, and (2) to minimize the error on the delay or distance estimation.

The number of nodes is in our case limited by the number of names we can extract from the DNS traces and the reachability of these names. In Figure 1 it can be seen that DNS only resolved 85.54% of the names. Of course, without having an IP address, we cannot use them in the topology since we cannot estimate their position. Such non-resolvable names stem from users providing non-existing names in address fields of browsers or outdated links in the Internet. Another source is the standard behavior of many browsers to append the local domain to relative names that could not be resolved. Hence, if a name cannot be resolved, it often appears twice in the DNS trace. Sometimes browsers also try to resolve IPv6 addresses if the local system supports this protocol stack. However, as we explored in [11], this has hardly any chance to succeed.

Another thing that can be seen in Figure 1 is that DNS problems cannot be the only reason of nodes been excluded from the topology, since there are altogether only 65.89% of the nodes are resolved and responding to ICMP pings.

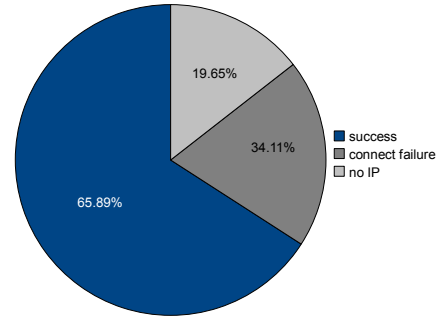


Figure 1. success rate of node name resolution and measurement

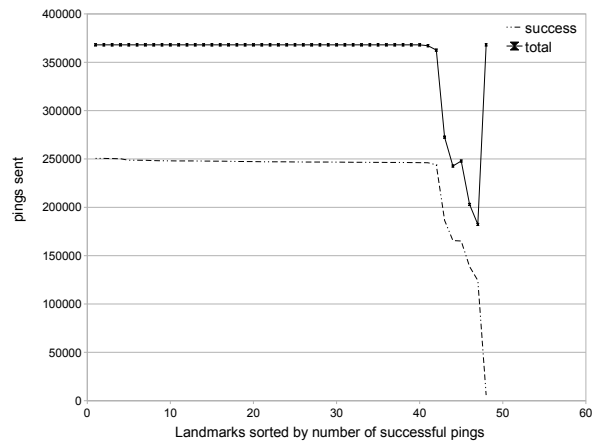


Figure 2. Number of pings send by each landmark vs. successful pings

However, even out of this fraction not all nodes are usable, since the nodes would have to respond to all landmarks, which is often not the case.

Please note that we exclude any names of e-mail black-lists and search engines that are provided in the case of DNS failures. We have seen many samples in our DNS traces, where spam filters query DNS to search a black list for an e-mail sender's host name. Therefore, we exclude all names that resolved to local IP addresses, which are usually answers from spam black lists or otherwise not usable anyway. Furthermore, we had to exclude the OpenDNS search page, which was often returned when otherwise an NXDOMAIN error would have been returned by a standard DNS server. These cases are included in the failures in Figure 1.

In Figure 2 one can see, how many pings were sent by each landmark and how many of them returned a result. We have used a trace with about 368,000 names. Even the best Planet-Lab nodes were only able to resolve about 250,000. Some hosts were so overloaded or unavailable that they were not even able to try all names. Note that we used

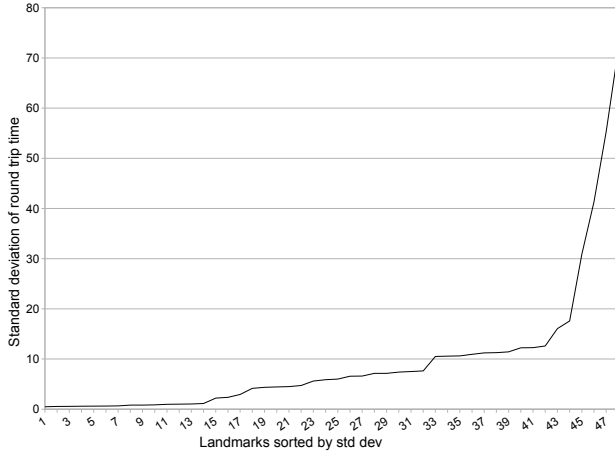


Figure 3. standard deviation of landmark measurements

CoMon to select Planet-Lab nodes without any problems for our selection at the beginning. An interesting observation is that some of the nodes had hardly any success even when trying all names. We mentioned the reasons for this phenomenon in Section 3.

Besides the number of successful pings, we also used the standard deviation of ping measurements as a fitness function to exclude certain landmarks, even if the measured many distances successfully. In Figure 3 one can see the measured distribution of the standard deviation. One can see that the deviation grows continuously until about 15 and then grows significantly. We used this threshold and excluded all nodes with a standard deviation of mode than 15, which is a very defensive value given the fluctuation of round trip times in the Internet and an extraordinary defensive value given the load variations on Planet-Lab nodes [15].

Trying to minimize the error of the pairwise latency estimation also influences the number of nodes available for the topology. Note that we use a notation of landmarks and dimensions to name our experiments, e.g., 3d4l means that an Euclidian space with 3 dimensions is used and 4 landmarks to estimate the coordinates.

In Figure 4 one can see how many nodes were usable for a topology with different numbers of dimensions and landmarks. Some nodes, that we called *outliers* have shown a deviation between calculated and measured distance to the landmarks of more then 10%. These nodes are not used later on. We are convinced that this threshold is selected very conservatively, given the deviations we have seen in our distance measurements.

The overall trend is that the number of usable nodes decreases with growing number of landmarks. The reason is, that the probability of a node being measured from all selected landmarks decreases with each additional landmark

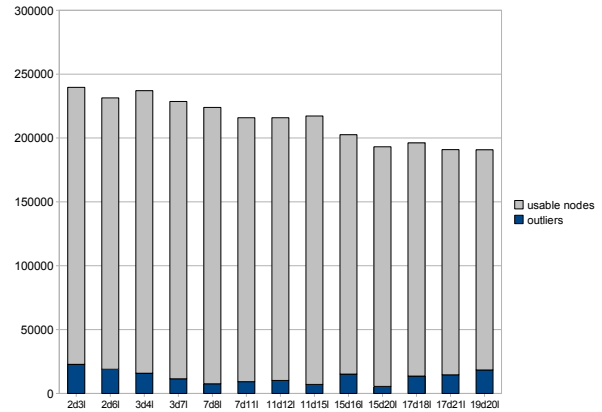


Figure 4. Nodes measured with sufficient precision vs. outliers in total numbers

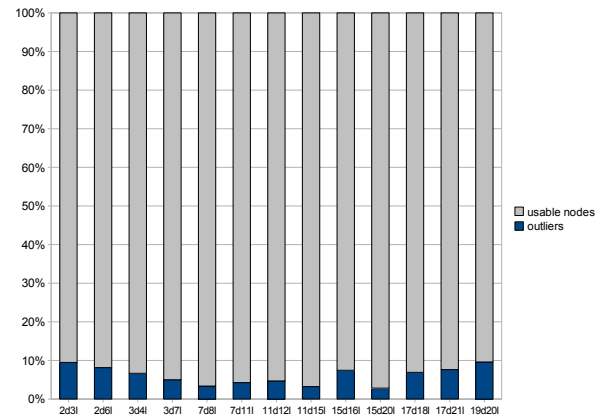


Figure 5. Nodes measured with sufficient precision vs. outliers

(see Figure 2). This has an even stronger influence when the number of landmarks is very small.

Nevertheless, sometimes it can be found, that with a higher number of landmarks even more nodes are usable, e.g. with 11 and 15 dimensions. The reason is, that with more landmarks the coordinate calculation can be improved. Hence, the number of outliers decreases. The relative amount of outliers among the nodes can be seen in Figure 5.

To achieve a good quality of the topology, as mentioned before, we have to keep the error in the calculated distances low. We compared the pairwise error in the distances between landmarks for different configurations of dimensions and landmark numbers to find out, what is a reasonable choice. The trade-off here is that the higher number of dimensions reduces the performance of the distance calcula-

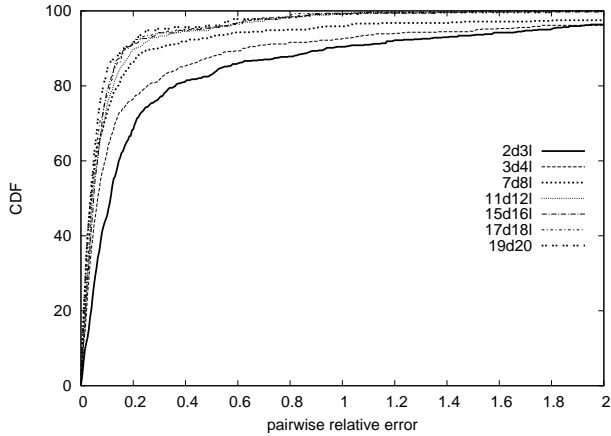


Figure 6. Pairwise relative error for $d+1$ landmarks

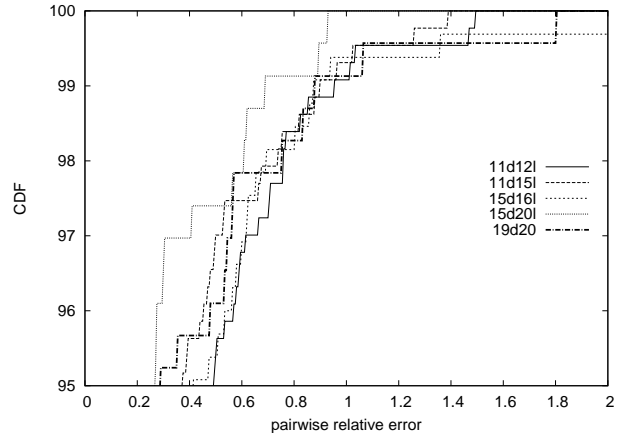


Figure 8. Pairwise relative error of the most precise solutions

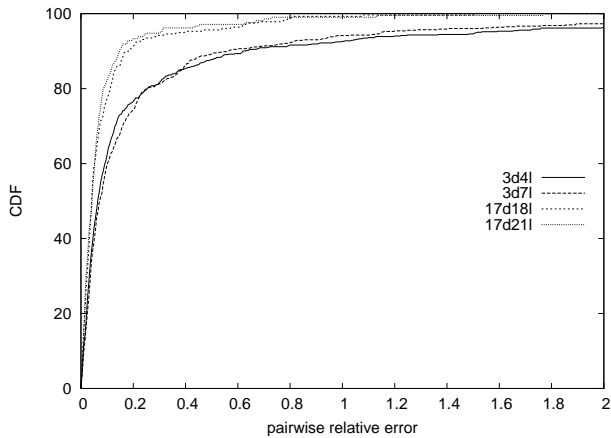


Figure 7. Pairwise relative error for different numbers of landmarks

tions while it offers more precision. Even if the additional cost for more dimensions was not important for us, it turns out more than 15 dimensions are not contributing any improvement anymore as can be seen in Figure 6. It shows the CDF of the pair-wise link delay estimation error. It can clearly be seen that 2 or 3 dimensions are not providing a good latency estimation.

We also tried to decide, whether it would be good to have more landmarks than $d + 1$ for d dimensions. In Figure 7 it can be seen that a better precision can be reached, if more landmarks are used. However, this benefit is not as significant as could be expected since the availability of high quality landmarks is limited. We sorted the landmarks to be used by number of measurements and standard deviation. Hence, with every additional landmark, the overall quality of landmarks decreases and so the overall increase of the topology's precision is rather limited.

In Figure 6 one can see a comparison between different numbers of dimensions d where we always selected $d + 1$ landmarks. It is obvious that for two and three dimensions there is a significant error. Approximately 10% of the distances have an error of at least 100%, while for more than 15 dimensions the amount of computed distances with such a deviation is negligible. With seven dimensions, we reached much better results. 72.91% of the computed distances had less than 10% error and only 4.1% had more than 100% error.

One can also see that we reached very similar results with 15, 17, and 19 dimensions. Between 71.86% and 84.42% of the distances had an error of less than 10%. With 15 dimensions and 20 landmarks we even had no computed distances which had more than 92.5% error. Note that we never created any topology with more than 21 landmarks due to the restricted availability of good landmarks.

In Figure 8 one can see that for an equal number of dimensions, more landmarks are an advantage. However, having more dimensions is not an advantage in general. With 15 dimensions and 20 landmarks, we see the best result, while if we would only have 16 landmarks, it would be better to reduce the number of dimensions. One can see that with 11 dimensions we reached better results than with 16 landmarks for 15 dimensions.

We also tried to find hints to be able to create better DNS name space-based transit stub topologies as proposed in [10]. Therefore, we selected three times 500,000 links and classified them based on the names of the two peers to be intra-stub (IS) for names within the same 2^{nd} -level domain, stub-transit (ST) for names in the same top level domain, or intra-transit (IT) otherwise.

As one can see in Table 1, there is always a significant standard deviation. However, the median suggests that similar names have most of the time a shorter distance to each

Statistics (ms)	IS	ST	IT
mean	50.76	86.90	110.82
standard deviation	74.65	62.57	72.54
median	10.40	75.72	112.29

Table 1. measured statistics for DNS name-based transit-stub-topologies

other. The mean however indicates that there are outliers with high link delay in the IS and TS statistics. For the IT links the mean is even smaller than the median, which indicates that there are some very small link delays decreasing it. It is however difficult to represent in a transit-stub-topology that some nodes in different stubs are tightly connected.

5. Conclusions

We use DNS traces and Planet-Lab nodes to generate realistic Internet topologies using TopDNS. There are several critical points where wrong assumptions can lead to poor precision in the generated topology.

When using DNS traces as sources to learn node names used in the Internet, it is not possible to include all nodes found in the trace. The reasons are manifold, but sometimes only 50% are included in the topology.

When using Planet-Lab landmarks, it is important to know that their workload sometimes is unstable and therefore, the precision of round trip time measurements is decreased. Also uptime is a critical point here. Planet-Lab nodes cannot be expected to take measurements over a long period of time. Hence, landmarks have to be sorted out using different criteria before using these measurements for coordinate calculation.

The number of dimensions is also a critical choice. TopDNS works best with 15 dimensions. We cannot exclude that it would work even better with more dimensions, if more high precision landmarks would be available.

Creating good transit-stub-topologies is still challenging and needs further research. However, realistic Internet topologies as TopDNS is creating them have many advantages in comparison to transit-stub-topologies. They represent a part of the real Internet. Results of simulations on such topologies offer a much more direct insight in the behavior of a system in the Internet. Furthermore, network information, like host names and real IP addresses, are already available in the simulation.

References

[1] K. Calvert, M. Doar, and E. Zegura. Modeling internet topology. *Communications Magazine, IEEE*, 35(6):160–163, Jun 1997.

[2] R. Cox, A. Muthitacharoen, and R. Morris. Serving DNS using a peer-to-peer lookup service. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 155–165, London, UK, 2002. Springer-Verlag.

[3] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: a decentralized network coordinate system. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 15–26, New York, NY, USA, 2004. ACM Press.

[4] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang. IDMaps: a global internet host distance estimation service. *IEEE/ACM Trans. Netw.*, 9(5):525–540, 2001.

[5] L. Garces-Erice, K. Ross, E. Biersack, P. Felber, and G. Urvoy-Keller. Topology-centric lookup service. Proceedings of the 5th International Workshop on Networked Group Communications (NGC'03), 2003.

[6] K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: estimating latency between arbitrary internet end hosts. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement*, pages 5–18, New York, NY, USA, 2002. ACM Press.

[7] T. S. E. Ng and H. Zhang. Predicting internet network distance with coordinates-based approaches. In *Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, volume 1, pages 170–179 vol.1, 2002.

[8] K. Park and V. S. Pai. Comon: a mostly-scalable monitoring system for planetlab. *SIGOPS Oper. Syst. Rev.*, 40(1):65–74, 2006.

[9] K. Park, V. S. Pai, L. L. Peterson, and Z. Wang. CoDNS: Improving DNS Performance and Reliability via Cooperative Lookups. In *OSDI*, pages 199–214, 2004.

[10] G. Pfeifer, C. Fetzer, and T. Hohnstein. Exploiting host name locality for reduced stretch p2p routing. In *6th IEEE International Symposium on Network Computing and Architectures (IEEE NCA07)*, July 2007.

[11] G. Pfeifer, A. Martin, and C. Fetzer. Reducible complexity in dns. In *Proceedings of the IADIS International Conference WWW/Internet 2008*, 2008.

[12] T. Roscoe. *The PlanetLab Platform*, chapter 33. The Planet-Lab Platform, pages 567 – 581. Lecture Notes in Computer Science Springer-Verlag GmbH, 2005.

[13] S. Srinivasan and E. Zegura. M-coop: A scalable infrastructure for network measurement. In *WIAPP '03: Proceedings of the The Third IEEE Workshop on Internet Applications*, page 35, Washington, DC, USA, 2003. IEEE Computer Society.

[14] W. Theilmann and K. Rothermel. Dynamic distance maps of the internet. In *In IEEE INFOCOM 2000, Tel Aviv*, 2000.

[15] T. Warns, C. Storm, and W. Hasselbring. Availability of globally distributed nodes: An empirical evaluation. In *Proceedings of the 27th Symposium on Reliable Distributed Systems (SRDS '08)*. IEEE Computer Society Press, 2008.

[16] E. Zegura, K. Calvert, and S. Bhattacharjee. How to model an internetwork. *INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, 2:594–602 vol.2, Mar 1996.