Locality Prediction for Oblivious Clients

Kevin P. Shanahan, Michael J. Freedman
New York University
www.coralcdn.org

Abstract

To improve performance, large-scale Internet systems require clients to access nearby servers. While centralized systems can leverage static topology maps for rough network distances, fully-decentralized systems have turned to active probing and network coordinate algorithms to scalably predict inter-host latencies. Internet applications seeking immediate adoption, however, must inter-operate with unmodified clients running existing protocols such as HTTP and DNS.

This paper explores a variety of active probing algorithms for locality prediction. Upon receiving an external client request, peers within a decentralized system are able to quickly estimate nearby servers, using a minimum of probes from multiple vantages. We find that, while network coordinates may play an important role in scalably choosing effective vantage points, they are not directly useful for predicting a client's nearest servers.

1 Introduction

Many replicated Internet systems can improve performance by servicing clients at nearby hosts. The performance of a distributed web mirror, for example, is highly dependent on the network distance between client and server. Commercial content distribution networks like Akamai [1] build large maps of the Internet topology and use this information to redirect clients to nearby hosts. These hosts are carefully deployed at specific access sites or behind bottleneck links. This technique for locality prediction, however, requires centralized mapping, aggregation, extensive network knowledge, and often ISP-specific heuristics. But by using existing protocols like HTTP and DNS, these systems can achieve immediate and widespread use.

More recent distributed systems use self-organizing techniques to reduce the infrastructure's administrative and operational overhead, while still providing service to unmodified clients. Such peer-to-peer systems include static and dynamic content distribution networks [5, 6], distributed hash storage services [7], and new Internet naming systems [13, 14]. Such decentralized systems, however, cannot easily produce aggregated static topology maps nor specify host deployments [4].

Active probing provides a simple alternative to static topology mapping that can be readily realized in a decentralized system. In its simplest form, when an unmodified client contacts any system peer, this ingress peer probes the client and directs other so-called *landmarks* to do the same. By collecting these round-trip-time (RTT) measurements, the ingress concludes which corresponding peer is closest to the client in terms of network distance, requiring no *a priori* knowledge of a client's location. Coupled with some application-level mechanism such as DNS redirection, this approach can be leveraged to service clients from nearby hosts.

Recent network coordinate systems [8, 10, 12, 9, 2, 3] offer new methodologies for active probing. These systems allow peers to estimate inter-host latency without topological maps or explicit all-pairs measurements, by assigning synthetic coordinates to each peer. The distance between peers in the coordinate space accurately predicts their RTT. Thus, for example, by combining active probing with network coordinates, a peer can map its client onto the system's underlying coordinate space. Then, it can use these coordinates to estimate the client's location and redirect it accordingly, potentially to a *destination* other than from among the landmarks.

This paper explores various methodologies for locality prediction using active probing. We concentrate our analysis on four main properties: (1) the method used for *landmark selection*, (2) the *number of landmarks* selected, (3) the method used for *destination determination*, and (4) the *number of redirection iterations* performed by a client.

We present and analyze several algorithms for landmark selection, destination determination, and methods of iterative redirection. We find that network coordinates *enable* landmark-selection algorithms that yield better client redirection, but are not as useful in *directly* finding a client's nearby hosts via coordinate distance estimation.

2 Design

In an abstract model, we consider a network comprised of a core decentralized system and external clients. Internal system peers communicate with one another, sharing live-

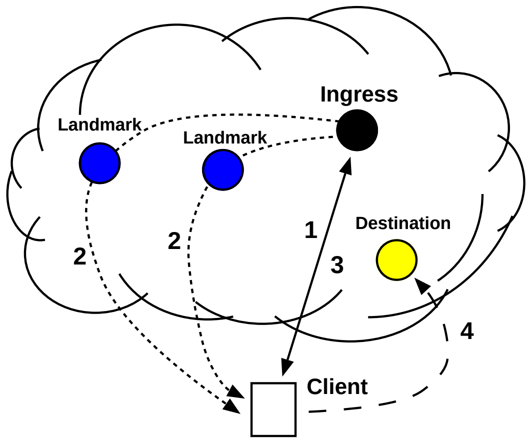

Figure 1: *Redirection architecture*. When an oblivious client c contacts any system peer (step 1), this ingress peer directs other known landmark peers to probe the client to determine RTT (step 2). The ingress accumulates these measurements and selects some destination peer that is closest to c. It returns this peer (step 3), which the client can subsequently contact for application-level operation (step 4). The destination may or may not be restricted to the set of landmarks (latter condition shown).

ness information and measuring internal round-trip-times, potentially as part of a network coordinate system.

We assume that an external client has some method of contacting a random system peer, via DNS for example.[1] As shown in Figure 1, upon receiving a request, this *ingress* peer selects some subset of system peers to act as *landmarks* for probing the client. All landmark peers probe the client after receiving a corresponding request from the ingress. Landmarks respond to the ingress with all measured RTTs to the client.

2.1 Network coordinates

In network coordinate systems, peers derive synthetic coordinates based on measurements to a limited subset of other peers, either using a fixed set of public landmarks [8, 10, 12] or in a fully-decentralized fashion [9, 2, 3]. Given another peer's coordinates, a peer can accurately predict its RTT without physical measurement.

In our decentralized model, hosts within the core peer-to-peer infrastructure can easily maintain and disseminate network coordinates as a by-product of other communication: (1) A peer includes its coordinates in all application-level messages it sends. (2) Whenever communication is two-way, the sender learns the RTT to the recipient using packet timestamps.

[1]For example, if the system is named with a particular domain and every peer runs a DNS server authoritative for the name, clients initially contact one of 13 or so peers listed with a registrar. Nameserver caching from prior requests can increase the client's view to the entire system.

For concreteness, we use the decentralized Vivaldi algorithm [3] for our network coordinate system. Vivaldi coordinates are represented by a (x, y, h, e) tuple, corresponding to a two-dimensional Euclidean coordinate space (x, y), with an additional height scalar h and an error term e. Conceptually, the Euclidean coordinates model the peer's location in the high-speed Internet backbone, the height models the additional access link latency at the Internet's edge, and the error term captures the host's confidence in its coordinates. The distance $D_{a,b} = \|C_a - C_b\|$ between two peers' coordinates C_a, C_b approximates their RTT; it is calculated by taking the Euclidean distance of their (x, y) coordinates and adding the height vectors. A peer updates its own coordinates whenever communicating with other peers, taking the others' error estimates into account.

External *unmodified* clients cannot themselves engage in the network coordinate algorithms. However, an ingress can estimate a client's coordinates, if desired, by collecting the landmarks' RTT measurements and coordinates. Then, it can synthesize client coordinates by running the centralized Vivaldi algorithm repeatedly on these measurements.[2]

2.2 Design considerations

The landmark selection and redirection mechanisms we explore rely on two assumptions. First, all internal peers can select a *random* subset of other peers from the network. Second, peers can obtain a subset of those peers closest to a specific distance from a particular peer.

The second assumption requires some practical consideration. Without using network coordinates, each peer must maintain complete $O(n^2)$ network state, and each peer must continually probe all others to generate these RTTs. Network coordinates, however, allow peers to predict all pair-wise latencies with only $O(n)$ state. In fact, each peer need only continually probe a small number of others, independent of n. To learn all n coordinates, a peer can then perform some gossiping protocol in an unstructured overlay. Alternatively, new distributed data structures may be designed to support closest-point queries (like a dynamic Voronoi diagram) using significantly less communication. We leave this as an open problem.

Based on having n^2 RTT estimates via n network coordinates, our results serve as an upper-bound for the performance benefit of more complex redirection algorithms over mere random selection. If less state is available at an ingress, this performance gap shrinks. System designers should therefore weigh this benefit in locality prediction against its usage and maintenance cost.

[2]The Vivaldi algorithm is analogous to a mass spring network, which does not converge immediately, but rather has a period of oscillation before resting at the (locally) lowest energy-level configuration.

System designers must consider other practical issues when performing active probing. To handle packet losses or excessive queuing delays, an ingress peer may choose to use $k + \alpha$ landmarks to ensure k timely results. Second, clients may be behind firewalls or NATs, where UDP probes, TCP requests to high ports, or ICMP packets have varied success. If client requests use connection-based protocols, RTTs can be measured directly during connection establishment. Or, fast traceroute-like scans can at least report timing information to the client's firewall. Still, the system should seek to use a minimal number of probes whenever possible, as additional probing increases network traffic, response time, and the probability of abuse complaints.

2.3 Selecting landmarks

We consider three different metrics for choosing k landmarks with which to probe the client.

Random. k peers are selected uniformly at random from the system, fresh for each query.

Well-distributed. This approach attempts to select landmarks that have good coverage throughout the network—*e.g.*, spread across North America, Europe, and Asia—without requiring static configuration. Thus, our algorithm works by selecting random subsets of k peers, then choosing the subset that minimizes the following:

$$|mean(\mathcal{D}_n) - mean(\mathcal{D}_k)|^2 + var(\mathcal{D}_k)$$

where $\mathcal{D}_k$ is the set of all pair-wise distances for the k peers in the subset (resp. n peers in the network).

Intuitively, minimizing the difference in mean distances ensures that landmarks are not clustered together. Minimizing variance ensures that peers are approximately equidistant from one another. Taken together, these properties attempt to spread landmarks evenly throughout the network. Experimental analysis (not included) shows that minimizing only one property yields strictly worse performance than that obtained using this given metric.

While this well-distributed metric is computationally more expensive, an ingress peer need not perform this selection process online nor upon each client request. For example, it may only reselect the set of well-distributed landmarks once every five minutes.

Sphere. In our third metric, the ingress attempts to choose landmarks that are likely to be closer to the client. Specifically, an ingress selects landmarks whose distances from the ingress are closest to the distance between ingress and client. This implies, of course, that the ingress must first calculate the RTT r to the client before choosing $k-1$ other such landmarks. We select $k-1$ non-ingress landmarks in order to fairly compare this metric with the former two, based on the total number of vantage points probing the client (k).

Intuitively, if one is working in a three-dimensional coordinate space, the resulting set of $k - 1$ landmarks and the client are located on the surface of a sphere with radius r, centered about the ingress peer. Thus, any particular landmark on this sphere is $\leq 2r$ from the client, provided the triangle inequality holds. While Vivaldi uses instead a 2-D space with a height vector, (x, y, h), we use this nomenclature for illustrative clarity.

Note that neither the sphere nor the well-distributed metrics strictly require network coordinates: The distance calculations used when selecting landmarks could be based on measured RTTs instead of coordinate distances, although this practice would significantly limit the system's scalability.

2.4 Selecting destinations

We consider two different metrics for determining the destination peer to which the client is redirected.

Direct RTT measurements. Given the k RTTs between landmarks and the client, based on direct network probes, the ingress peer chooses the landmark with the smallest measured RTT.

Estimated coordinate distance. After collecting the k RTT measurements from landmarks, as well as the landmarks' latest coordinates, the ingress peer synthesizes the client's coordinates C_c per Section 2.1. Then, given the coordinates of all other system peers, the ingress chooses the peer d that is closest to the client in the coordinate space, *i.e.*, minimizes $D_{c,d} = \|C_c - C_d\|$. Note that, unlike the direct measurement approach, this destination is not restricted to the set of landmarks.

2.5 Iterating redirection

Finally, we consider whether repetitions of the redirection mechanism will improve the system's predictive accuracy, where each step of the algorithm attempts to find a destination closer to the client.[3] Thus, the destination from iteration $i-1$ is contacted by the client as an ingress peer for iteration i. Although such iteration does not make sense for random landmark selection, we can consider iteration for the latter selection metrics.

For the *well-distributed* landmark metric, we attempt to decrease the mean distance between peers by half during each iteration, using $|2^{-i} \cdot mean(D_n) - mean(D_k)|^2 + var(D_k)$ for the ith iteration. The ingress peer *must* include itself among the landmark set for $i > 0$, given that

[3] An iterative redirection mechanism is readily feasible even for unmodified clients. For example, custom DNS servers can synthesize artificially-hierarchical hostnames, as in [5], causing DNS resolvers to resolve the names recursively.

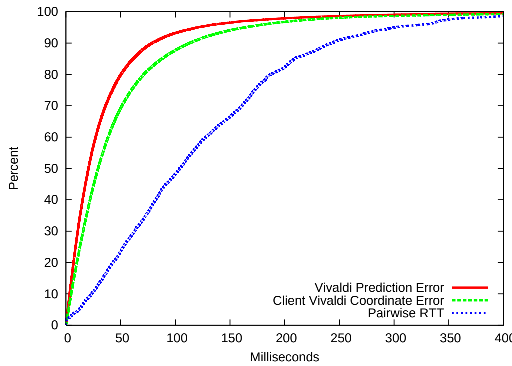

Figure 2: Coordinate predictive error vs. network RTTs

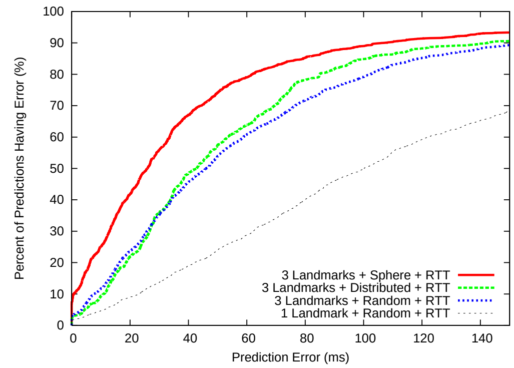

Figure 3: Comparison of landmark selection metrics

the algorithm should select some set of well-distributed peers that are closer to the client at each iteration.

For the *sphere* metric, each iteration proceeds as expected. Intuitively, the ingress from iteration i selects landmarks from a sphere with radius $r_i \leq r_{i-1}$. To ensure forward progress, landmarks should not be reused between iterations. As the ingress of iteration i can use its measured RTT to the client from $i-1$ whenever possible, it selects k other landmarks for $i > 0$, not $k-1$ as described for $i=0$ above.

3 Evaluation

This section evaluates the proposed selection metrics for choosing both landmarks and destinations, as well as the effect of the number of landmarks and of iterations.

3.1 Methodology and terminology

We performed wide-area experiments on the PlanetLab testbed [11] and then simulated client-system interactions. On 105 randomly-chosen PlanetLab hosts (as of November 2004), we ran peers that implemented the Vivaldi network coordinate algorithm [3], where each peer regularly probed 32 others. These peers functioned as landmarks to allow the collection of RTT measurements accompanied by their sources' network coordinates. Peers sent ICMP echo messages as probes; we used the response's kernel timestamp to minimize the effect of scheduling latency.

A non-PlanetLab server directed *all* peers to probe each client once, with a 25 ms delay between each probe request to reduce congestion at the specified client. These clients were restricted to the same set of PlanetLab peers, although peers did not simultaneously play the roll of client and landmark. Each peer was simulated as a client three times. We collected the resulting ~30,000 RTT measurements for subsequent analysis.

Figure 2 briefly characterizes the predictive error of our Vivaldi implementation, defined by the difference between predicted distance in network coordinate space and the measured RTT for any two peers a, b: $|D_{a,b} - RTT_{a,b}|$. This is plotted alongside the cumulative distribution function (CDF) of all measured pair-wise RTTs. Additionally, we plot the error of the client coordinates synthesized by the ingress using only 3 landmarks, against all peers as before. Thus, we find that client coordinates can achieve relatively good accuracy, even though these coordinates are "fitted" against only a few points. Using 32 landmarks yielded client results indistinguishable from that of internal peers.

For our analysis, we consider how effectively a mechanism can predict the system's *optimal* destination o. Given that our experiments calculate RTTs between a client c and all system peers, we call the peer with minimum RTT optimal. We say that the metric's *predictive error* is the absolute RTT difference between the client with the predicted destination d and with the optimal peer: $|RTT_{c,d} - RTT_{c,o}|$.[4]

3.2 Results

We now evaluate the specified metrics and parameters for active probing. For predictive sphere selection, a random peer was selected as an ingress peer for each client test. For random landmark selection, a random subset was selected for each client test. For the distributed metric, 10,000 subsets were considered, chosen uniformly at random with replacement.[5] The subset that minimized the well-distributed metric (per Section 2.3) was used for the duration of the experiment. The following figures are the combined results from 10 such evaluations on all clients. As a baseline, each graph includes the error CDF of using one randomly-selected destination.

[4] We note that an alternate metric to consider is the relative RTT difference, or stretch, which normalizes the absolute difference by the optimal RTT. We do not include such analysis in this paper, however, due to our interest in the system's absolute performance: An increase from 2ms to 5ms, while having high stretch, is irrelevant in practice.

[5] Testing all $\binom{n}{k}$ possible subsets was computationally infeasible.

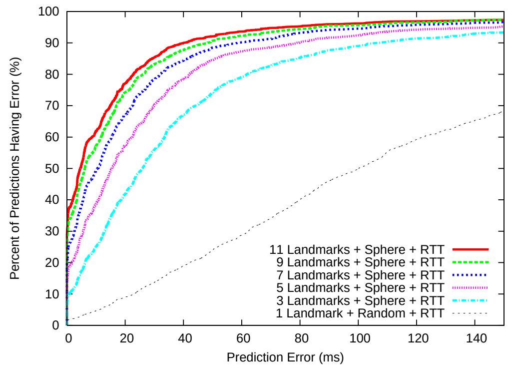

Figure 4: The effect of the multiple landmarks

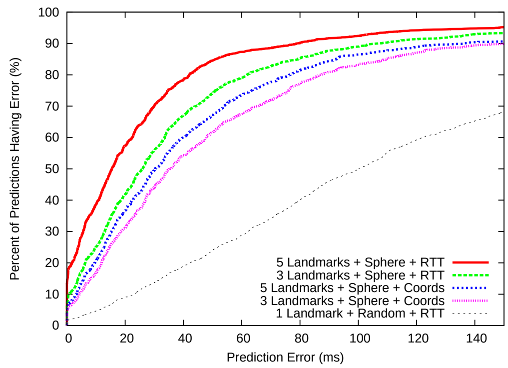

Figure 5: Comparison of destination selection metrics

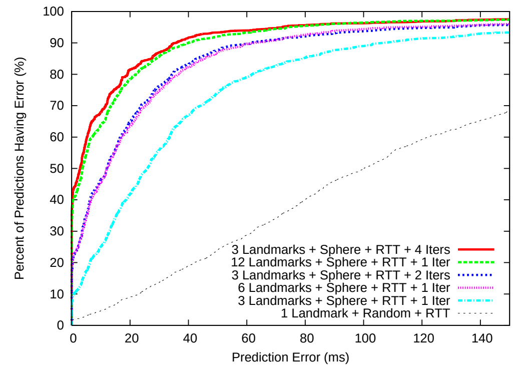

Figure 6: Redirection iteration vs. landmark number

Landmark selection metrics. Figure 3 compares the efficacy of the three landmark selection metrics for decreasing predictive error. All three CDFs shown use three landmarks and the direct RTT measurement approach for selecting a destination. An error of 0 ms corresponds to predicting the optimal destination. The sphere metric makes the most accurate predictions, with a median error of 25.9 ms and optimal selection of 6.8%. The distributed metric has a median error of 42.8 ms and optimal selection of 2.1%. Randomly selected landmarks have a median error of 45.9 ms and optimal selection of 2.3%.

Number of landmarks. Figure 4 shows the effect of multiple landmarks on predictive error. Increasing the number of landmarks improves the accuracy of destination selection, with decreasing returns as the set size increases. Thus, even a moderate number of landmarks greatly improves performance: Using 3 landmarks results in a destination with median predictive error that is 4x better than that of random selection (*i.e.*, 1 landmark).

Destination selection metrics. Figure 5 compares the two methods for selecting destinations, that of direct RTT measurement versus using network coordinates to estimate a client's location and predict a nearby peer. Somewhat surprisingly, we find that the coordinate prediction approach yields strictly worse performance. These results are shared across all landmark selection metrics and all evaluated landmark set sizes (up to 11).

A (strong) form of the destination selection problem is the following: generate a sorted list of peers with increasing distance from the client. The RTT metric samples from this list in some (possibly biased) manner. The coordinate metric uses the same set of peers, but sorts on coordinate distance. Therefore, any inaccuracy in the coordinate system—which is inherent whenever mapping the Internet onto a lower-dimensional space—is reflected in the differences between the orderings of these two lists and hence in their first elements. This intuition is reflected by the data: the median error in destination prediction when using coordinates is 35.5 ms, while the median error in the coordinates' accuracy themselves (Figure 2) is 29.4 ms. Lower-error coordinate systems can potentially improve the accuracy of destination selection based on network coordinates.

Number of iterations. Next, we examine the effect of iteratively repeating the redirection mechanism. Each iteration uses a set of k landmarks not included in previous iterations. We find that iteration improves accuracy, but again with diminishing results. The results are similar across all evaluated selection metrics (noting that random selection cannot be expressed as an iteration problem).

One must consider whether these benefits are simply caused by increasing the total number of landmarks, instead of any additional "forward progress" made during each iteration step. Figure 6 compares the predictive error of k landmarks against i iterations of j landmarks each, where $i \cdot j = k$, again using the sphere and RTT metrics. We find that iterations do have some additional benefit beyond merely increasing the number of landmarks, although it is quite small. Thus, system designers must weigh whether the gain from iteration (in returning a closer destination) outweighs its cost (in added latency during the redirection protocol) for their particular application domain.

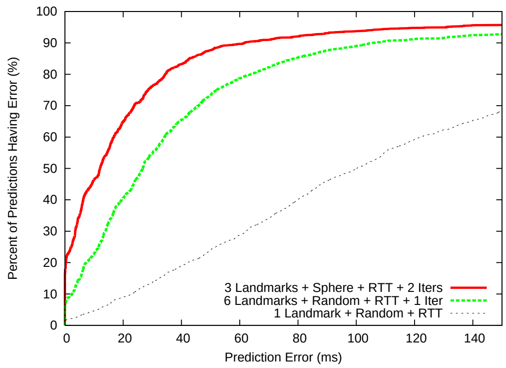

Figure 7: Simple random vs. iterative distance metrics

Repeated probing. Additionally, one might consider the effect of taking multiple probes versus using only a single probe per landmark, given that any transient network congestion can cause queuing delays and affect RTT measurements. However, we found no real difference in predictive errors between the two cases.

Minimizing complexity. In summary, Figure 7 shows the benefit of using the more complex locality prediction mechanisms explored in this paper. We compare the simple random approach to our "best" solution, which couples iteration with the sphere landmark metric. We see that 6 random landmarks achieve a median predictive error of 26.7 ms and a 90th percentile of 105.4 ms. Using two iterations of three sphere-chosen landmarks, we decrease the median error to 12.0 ms (a 2.2x improvement) and the 90th percentile to 61.5 ms (1.7x better). Certainly, both are better than random selection, with its median of 101.1 ms and 90th percentile of 230.5 ms. Thus, while the advantage of the more complex redirection mechanism is significant, we also conclude that the simplest approach can in fact achieve reasonable results.

4 Conclusions

To improve performance, large-scale Internet systems require clients to access nearby servers. Unfortunately, the techniques usually associated with generating static topology maps—centralized mapping, trustworthy aggregated results, extensive network knowledge, ISP-specific heuristics, infrastructure deployment, etc.—are not readily available to fully-decentralized systems. Thus, such systems have turned to active probing and network coordinate algorithms to scalably predict inter-host latencies.

This paper explores various methodologies for locality prediction using active probing, showing that a distributed system can achieve high accuracy with minimal probing. We concentrate on four properties. (1) We conclude that, when selecting landmarks, a peer should choose ones that are more likely to be close to the client (the so-called sphere metric). If such selection is not feasible, well-distributed landmarks perform better than randomly-selected peers, but only when using smaller numbers of landmarks. (2) Increasing the number of landmarks improves the system's predictive accuracy, although with diminishing returns. (3) Iteratively redirecting clients provides minimal improvements over that obtained only from increasing the number of landmarks. (4) Choosing a client's destination based on coordinate distances yields worse accuracy than simply using the landmark with smallest RTT. Thus, network coordinates enable certain landmark-selection algorithms—like the sphere metric—to scale to large systems, but are not as directly useful for determining a client's nearest hosts.

Acknowledgments. Special thanks to Russ Cox, Frank Dabek, Jinyang Li, David Mazières, and members of the NYU Systems Group for helpful discussions.

References

[1] Akamai Technologies. http://www.akamai.com/, 2004.

[2] M. Costa, M. Castro, A. Rowstron, and P. Key. PIC: Practical Internet coordinates for distance estimation. In *Conference on Distributed Systems*, Tokyo, Japan, March 2004.

[3] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *ACM SIGCOMM*, Portland, OR, August 2004.

[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. on Networking*, Oct 2001.

[5] M. J. Freedman, E. Freudenthal, and D. Mazières. Democratizing content publication with Coral. In *NSDI*, San Francisco, CA, March 2004.

[6] R. Grimm, A. Kravetz, G. Lichtman, N. Michalakis, S. Raza, A. Elliston, and J. Miller. OpenEdge: A unified architecture for edge-side content creation, transformation, and caching. Technical report, NYU, February 2005.

[7] B. Karp, S. Ratnasamy, S. Rhea, and S. Shenker. Spurring adoption of DHTs with OpenHash, a public DHT service. In *3rd Intl. Workshop on Peer-to-Peer Systems*, San Diego, CA, February 2004.

[8] E. Ng and H. Zhang. Predicting Internet network distance with coordinates-based approaches. In *IEEE INFOCOM*, New York, NY, June 2002.

[9] E. Ng and H. Zhang. A network positioning system for the Internet. In *USENIX Conference*, Boston, MA, March 2004.

[10] M. Pias, J. Crowcroft, S. Wilbur, S. Bhatti, and T. Harris. Lighthouses for scalable distributed location. In *2nd Intl. Workshop on Peer-to-Peer Systems*, February 2003.

[11] PlanetLab. http://www.planet-lab.org/, 2004.

[12] L. Tang and M. Crovella. Virtual landmarks for the Internet. In *Internet Measurement Conference*, Miami Beach, FL, October 2003.

[13] M. Walfish, H. Balakrishnan, and S. Shenker. Untangling the Web from DNS. In *NSDI*, San Francisco, CA, March 2004.

[14] M. Walfish, J. Stribling, M. Krohn, H. Balakrishnan, R. Morris, and S. Shenker. Middleboxes no longer considered harmful. In *OSDI*, San Francisco, CA, December 2004.