In Proceedings of IEEE ICDCS 2008.

Relative Network Positioning via CDN Redirections

Ao-Jan Su, David Choffnes, Fabián E. Bustamante and Aleksandar Kuzmanovic
Department of Electrical Engineering and Computer Science, Northwestern University
{ajsu,drchoffnes,fabianb,akuzma}@cs.northwestern.edu

***Abstract*—Many large-scale distributed systems can benefit from a service that allows them to select among alternative nodes based on their relative network positions. A variety of approaches propose new measurement infrastructures that attempt to scale this service to large numbers of nodes by reducing the amount of direct measurements to end hosts. In this paper, we introduce a new approach to relative network positioning that *eliminates* direct probing by leveraging *pre-existing* infrastructure. Specifically, we exploit the dynamic association of nodes with replica servers from large content distribution networks (CDNs) to determine relative position information – we call this approach CDN-based Relative network Positioning (CRP). We demonstrate how CRP can support two common examples of location information used by distributed applications: server selection and dynamic node clustering. After describing CRP in detail, we present results from an extensive wide-area evaluation that demonstrates its effectiveness.**

Keywords: Network positioning systems, content distribution networks, measurement reuse.

I. Introduction

Most wide-area networked systems, such as data sharing services [9], [37], overlay-based multicast [2], [7], [32], distributed games [3], [22] and content distribution networks [5], [44] could benefit from information regarding the relative proximity of participating hosts. For example, data sharing systems could select among replica servers based on their distance from a requesting client. Streaming multicast systems could optimize their overlays by structuring them based on relative distances between machines. Finally, distributed online games could balance server load while satisfying real-time delay constraints by organizing participants in clusters of nearby players.

Various methods have been proposed to support such a service in a scalable manner, without requiring the overhead of all-to-all measurements. These include using proxies [15], landmark binning [36], direct measurement [46] and decentralized network embedding (such as [12], [30], [31], [33], [41], [43]). With network embedding, for instance, synthetic coordinates in a geometric space are employed to characterize node locations, and network distances between nodes are estimated based on their corresponding vector distances.

Although the different proposed methods have attractive properties, recent studies have shown that they leave much room for improvement in terms of practicality and prediction accuracy, particularly in systems with high degree of churn [21], [26], [46], [47]. For example, while network embedding ensures scalability by avoiding direct measurements, the embedding process itself can introduce significant errors (e.g. in the selection of landmarks). Although an structured approach to direct measurement can avoid some of these issues by no relying on an absolute coordinate space, it achieves this by re-introducing direct measurement and its accuracy strongly depends on the time available for on-demand probing [46].

Many distributed applications, however, do not require exact topological information and could instead build on sufficiently precise hints about the relative position of networked hosts. Server selection in an on-line gaming system and binning of peers for overlay construction are clear examples. For these applications, relative order is more important than absolute distances [36]. In this paper, we introduce a new, practical approach to relative network positioning that *eliminates* direct probing by leveraging the dynamic association of hosts with replica servers from large content distribution networks (CDNs). We call this approach *CDN-based Relative network Positioning (CRP)*.

CDNs cache copies of web objects on thousands of replica servers worldwide [1] and redirect web clients to relatively small sets of different replica servers over time. In [42], we demonstrated that such *redirections are primarily driven by network conditions and updated frequently enough as to be useful for control.* CRP is based on the hypothesis that if two hosts see the same (or similar) set of nearby replica servers over time, they are likely to be relatively close to each other. Thus, CRP can estimate relative distances between hosts by comparing the set of replica servers to which they are redirected.

CRP provides a lightweight and highly scalable approach to relative network positioning. By relying on the network views collected by large-scale CDNs, CRP offers accuracy comparable to that of alternative approaches while avoiding *additional* direct measurements either to landmarks or to other peers in an overlay. Further, because it uses a well-known interface to existing DNS infrastructure, CRP is immediately available and easy to integrate in existing applications. We show that maintaining CRP redirection information at each node is highly scalable, requiring only infrequent requests *independent of the number of nodes using the system* ($O(1)$).

We argue that a CRP-based service can be commensalistic with CDNs (i.e. not harm the CDNs it relies on) and posit that it can even form the basis of a new, mutualistic service. Still, CRP is *not* intended as a general solution to the network positioning problem – if two hosts are never redirected to common replica servers, CRP is of no use in estimating their relative positions and can only indicate that the two nodes are *not* near one another. Nevertheless, in many location problems in distributed systems, the most useful information is precisely that which CRP can provide.

In this paper, we make the following contributions:

- We introduce CDN-based Relative network Positioning

(CRP), a lightweight, scalable and accurate approach to relative network positioning.
- We describe the benefits of CRP in the context of two common uses of location information by distributed applications: server selection and dynamic node clustering.
- We present results from an extensive evaluation of the effectiveness of the CRP approach based on large-scale measurements with over 1,000 hosts distributed worldwide.
- We explore the potential impact of a CRP-based service on CDNs, and discuss viable models of interaction between such a service and CDNs.

The remainder of this paper is organized as follows. We review related work in Section II. Section III briefly describes how CDNs work before introducing our CDN-based approach to relative network positioning. We discuss the use of CRP to support two common uses of location information in Section IV and report results from our wide-area evaluation in Section V. We discuss additional issues related to our CDN-based approach, including its potential costs, in Section VI and conclude in Section VII.

II. Related Work

CRP is the first approach to network positioning based on strategic reuse of CDNs' network measurements. The following paragraphs set the context for our work by briefly reviewing past CDN-related studies and surveying approaches to scalable network distance estimation and, more generally, "information plane" services for globally-distributed systems.

CRP leverages the dynamic association of Internet hosts with CDNs' replica servers. Previous work has analyzed the effectiveness and impact of CDNs [13], [18], [20], [39]. In [19] the authors examine how content distribution servers improve latency when compared to throughput from the origin servers. Based on a study of CDN redirection from two different CDN providers, Johnson et al. [16] argue that these CDNs commonly avoid bad recommendations rather than select optimal servers. More recently, through a detailed measurement of the Akamai CDN, we show that CDN redirections are primarily driven by network conditions, specifically network latencies on the paths between clients and the Akamai servers, and are updated frequently enough as to be useful for control [42]. Our early study illustrated the potential benefits of employing CDN redirections for identifying good detouring paths and demonstrated that in approximately 50% of scenarios, the best measured one-hop path through an Akamai server outperforms the direct path in term of latency.

There has been a variety of proposed approaches for supporting accurate network distance estimation. IDMaps [12] is an early service that estimates latency between arbitrary pairs of nodes using a small set of strategically placed tracer nodes. These tracer nodes proactively measure distances among themselves and representative nodes from each address prefix, then use these distances to generate a virtual distance map of the Internet. IDMaps depends on the deployment of a system-wide infrastructure and incurs errors based on the distances between clients and their closest tracers. Chen et al. [6] propose an approach that avoids the need for topology knowledge by clustering nodes based on latency and selecting node leaders to carry inter- and intra-cluster measurements and to respond to latency queries. The accuracy of their approach depends on how amenable the network is to clustering and its overhead is proportional to the number and size of the resulting clusters.

More recent approaches use synthetic coordinates in a geometric space to characterize node locations and compute distance estimates. Ng and Zhang [30] show that Internet distances can be embedded in a low-dimensional Euclidean space, and the network latency between two nodes can be estimated based on their network coordinates. These network coordinates are computed from distances to a set of landmarks or via a simulation-based approach, where coordinates are modeled as entities in a physical system (e.g., massless bodies in a spring relaxation problem). ICS [24] and Virtual Landmarks [43] first assign coordinates based on the distances to landmarks before applying principal component analysis to reduce the dimensionality of the coordinates. In [10], [23] landmarks are used only for bootstrapping and node coordinates are then computed based on the coordinates of peers. In systems with high degrees of churn, this could result in compounded embedding errors over time. Based on GNP [30], NPS [31] builds a hierarchical architecture to ensure convergence. Lighthouse [33] avoids fixed landmarks and relies instead on nodes already in the system to obtain a coordinate relative to them, which are then converted into a global coordinate by solving a system of linear equations. Rather than relying on landmarks, simulation-based systems compute coordinates based on the modeling of physical systems. Vivaldi [11] uses spring relaxation while Shavitt et al. [41] models a potential force field instead.

The above approaches require extensive latency measurements to estimate absolute network positions. Our focus is instead on supporting a relative network positioning system as that proposed by Ratnasamy et al. [36], but without requiring landmark selection or *additional* measurements. Gummadi et al. [15] proposes to leverage the existing DNS infrastructure and estimate the latency between two nodes as the measured latency between their DNS servers. Like King [15], CRP is easy to deploy and use as it leverages existing CDN infrastructures and provides a well-known interface that simplifies application integration.

CRP is not intended as the basis of a general latency estimation system, but as a lightweight approach to solve relative network positioning problems commonly found in distributed systems. Meridian [46] also solves spatial queries, without relying on a virtual coordinate system, building instead on direct measurements and a loosely structured overlay network. A Meridian node keeps track of a small fixed number of other nodes in the system organized into a set of concentric, non-overlapping rings. To promote geographic diversity in ring members, Meridian nodes periodically reasses ring-membership decisions with the goal of maximizing the hypervolume of the polytope formed by the selected nodes. For node discovery and dissemination Meridian relies on a simple gossiping mechanism based on an anti-entropy push protocol. While Meridian's direct-measurement approach can avoid some of the issues with coordinate-based systems, it does this by re-introducing direct probing and its accuracy strongly depends on the time available for on-demand measurement [46].

More generally, a number of research efforts have begun to address some of the challenges in supporting Clark et al.'s [8] grand vision of a knowledge plane for large-scale, self-managing distributed systems. Examples projects that address the scalable monitoring of end-hosts and network paths, and the efficient support of query processing in the information plane include Sofia [45], IrisNet [14], PIER [35], Underlays [29] and iPlane [27]. A CRP-based service complements these efforts providing a practical, easy-to-use, highly scalable approach to relative network positioning.

III. CDN-Based Relative Network Positioning

We begin this section with a short background on CDN operations before describing our CRP approach to network positioning in more detail.

A. Content Distribution Networks

CDNs attempt to improve web performance by delivering content to end users from multiple, geographically dispersed servers located at the edge of the network [1], [25], [28]. Content providers contract with CDNs to host and distribute their content. Since most CDNs have servers in ISP points of presence, clients' requests can be dynamically forwarded to topologically proximate replicas. DNS redirection and URL rewriting are two of the commonly used techniques for directing client requests to a particular server [17], [40].

Beyond static information, such as geographic location and network connectivity, most CDNs rely on network measurement subsystems to incorporate dynamic network information in replica selection and determine high-speed Internet paths over which to transfer content within the network [4]. Among CDNs, Akamai's is perhaps the most extensive distribution system in the world with over 25,000 servers, operating in approximately 1,000 networks.

B. CDN-Based Relative Network Positioning

CDN-based relative network positioning (CRP) is based on the hypothesis that one can estimate relative distances among hosts by comparing their respective sets of CDN replica servers with which these hosts are associated over time.

Each host keeps track of the servers to which a CDN redirects it over time. Each set of redirections can be compactly represented as a map of ratios, where each ratio represents the frequency with which the host has been directed toward the corresponding replica server during the past time window. Specifically, if node A is redirected toward replica server r_1 30% of the time and toward replica server r_2 70% of the time, then the corresponding ratio map is:

$$\nu_A = \langle r_1 \Rightarrow 0.3, r_2 \Rightarrow 0.7 \rangle$$

More generally, the ratio map for a node N is a set of *(replica-server, ratio)* tuples represented as

$$\nu_N = \langle (r_k, f_k), (r_l, f_l), ..., (r_m, f_m) \rangle$$

For brevity, we use $\nu_{N,i}$ to represent the ratio of time f_i that node N is redirected to replica server r_i. Note that each node's ratio map contains only as many entries as replica servers seen by that node and that the sum of the f_i's in any given ratio map equals one. Despite the large number of replica servers world-wide, in our study we have found that hosts see a small set of replica servers (< 20) very frequently and see others much less so.

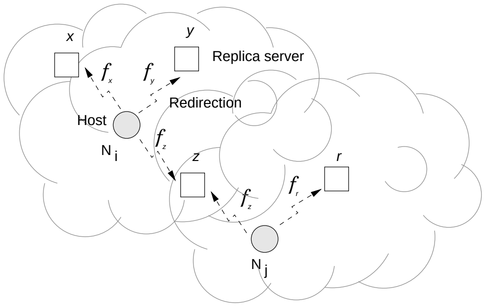

Fig. 1: Hosts are associated with different CDN replica servers over time. Each host keeps track of its redirection frequencies.

Based on our formulation of ratio maps, each node in our overlay can be represented as a vertex in a general graph connected by edges labeled with the degree of overlap in their redirection frequency maps. Following from the premise that CDN redirections are primarily driven by network conditions, the structure of this graph can be used to determine the relative position of hosts based on the *cosine similarity* of their ratio maps.

Cosine similarity [38] is a mathematical measure, with values on a scale of $[0, 1]$, of how *similar* two vectors are. Given two hosts A and B and treating a redirection map as a vector, the cosine similarity between hosts A and B can be formally defined as:

$$cos\_sim(A, B) = \frac{\sum_{i \in I_A} (\nu_{A,i} \times \nu_{B,i})}{\sqrt{\sum_{i \in I_A} \nu_{A,i}^2 \times \sum_{i \in I_B} \nu_{B,i}^2}}$$

Where I_A represents the set of replica servers to which node A has been redirected over the time window. Intuitively, the cosine similarity metric is analogous to taking the dot product of two vectors and normalizing the result. When the maps are identical, their resulting cosine similarity value is 1; when they are orthogonal (i.e., the hosts have no replica servers in common), the value is 0.

Thus, to determine the relative position of two hosts A and B with respect to a third host C, we can simply compute the cosine similarity of their respective redirection maps. In particular, if $cos\_sim(A, C) < cos\_sim(B, C)$, then host A is the closer to C of A, B.

Note that CRP is not, nor is intended as, a general solution to the network positioning problem – if two hosts are never redirected to common replica servers (e.g. a host located in Buenos Aires and one placed in Delhi), CRP is of no use in estimating their relative network positions. Stated more formally, CRP cannot determine proximity between two nodes if the dot product of their redirection maps is zero. In such a case, CRP can only indicate that the two nodes are not likely to be near one another, assuming that the CDN used for positioning has sufficiently broad coverage. CRP is not a complete service, but an approach that can serve as the basis for a lightweight and highly scalable service that supports common uses of location information in distributed systems.

In this paper, we focus on the *feasibility* of reusing CDN information to determine relative position information among nodes and leave the implementation of this approach as future work. We note, however, that a CRP-based service could be easily built as a stand-alone service, shared by multiple applications, or as part of an application library that takes advantage of application-specific communication to distribute redirection maps [34].

IV. Uses of CDN-Based Relative Network Positioning

CRP provides a general approach to solve some common relative positioning problems found in distributed systems. In the following sections, we describe two of its potential uses: closest node selection and node clustering.

A. Closest Node Selection

One of the most common uses of location information in a distributed system is for identifying nodes near a target host. In particular, it is often desirable to select the closest (i.e., lowest-latency) server host to a particular client host in the system. For example, interactive massively multi-player online games could use location information to improve latencies by assigning clients to nearby hosts in their mirrored server architectures. Similarly, peer-to-peer data sharing networks could optimize response time by downloading from nearby servers.

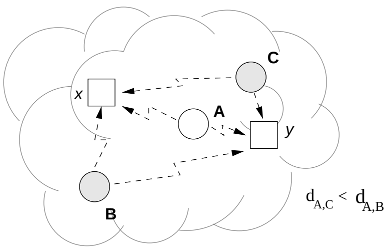

Fig. 2: CRP can support closest node selection, allowing client node A to select between servers B and C by comparing its cosine similarities for each of the potential servers.

CRP supports closest node selection based on the comparison of cosine similarities between the client host and each of the potential servers. Figure 2 illustrates CRP-based closest node selection with a small example of three nodes, where node A is the client host selecting between servers B and C. All three nodes are dynamically redirected to CDN replica servers x and y with varying frequency. For example, assume the nodes generate the following ratio maps:

$$\nu_A = \langle r_x \Rightarrow 0.2, r_y \Rightarrow 0.8 \rangle$$

$$\nu_B = \langle r_x \Rightarrow 0.6, r_y \Rightarrow 0.4 \rangle$$

$$\nu_C = \langle r_x \Rightarrow 0.1, r_y \Rightarrow 0.9 \rangle$$

As the following simple calculations show, the cosine similarity between the pair (A, C) is higher than that of (A, B), so node A selects server C.

$$cos\_sim(A,B) = \frac{0.2 \times 0.6 + 0.8 \times 0.4}{\sqrt{(0.2^2 + 0.8^2) \times (0.6^2 + 0.4^2)}} = 0.740$$

$$cos\_sim(A,C) = \frac{0.2 \times 0.1 + 0.8 \times 0.9}{\sqrt{(0.2^2 + 0.8^2) \times (0.1^2 + 0.9^2)}} = 0.991$$

B. Clustering

Another application of CDN-based network positioning is node clustering, where the objective is to divide a set of peers into disjoint groups (i.e., clusters) according to the specified constraints, such as cluster diameter or the average distance from the "center" of the cluster. In our clustering technique, we define the cluster distance to be in terms of round-trip time (RTT) latency and our objective is to find clusters such that each node in each cluster is closer to the center of its cluster than to any other cluster center.

In general, a clustering service should be able to address the following queries:

- *Given a node identifier, find the other nodes that belong to the same cluster.* This is useful, for example, in swarming peer-to-peer systems (such as BitTorrent) where a node wishes to peer with nodes on low RTT paths so as to minimize latency and potentially increase bandwidth.
- *Given a set of nodes, map each node to a cluster.* In an overlay network, a quality overlay path may become unavailable due to churn. When a node along a path goes down, one can use knowledge of clusters to quickly repair the path and maintain its quality by using another node in the same cluster.
- *Given a set of m nodes, find n ($\leq m$) nodes in different clusters.* Nodes in different clusters are often in different regions of the Internet. The answer to the preceding question allows one to find a group of peers for which network faults are not correlated (with high probability), an essential service for systems providing high reliability.

Figure 3 provides a high-level illustration of how our CRP-based clustering service works. As usual, nodes monitor their redirections toward replica servers. In the figure, nodes A, B and C are all directed to replica server v and nodes D, E and F are all directed toward replica server x. Because these two sets of nodes do not otherwise have replica servers in common, we form clusters $\{A, B, C\}$ and $\{D, E, F\}$.

More generally, we use the cosine similarity between two nodes' redirection behavior as the distance metric for our clustering algorithm. Recall that the main hypothesis driving our CDN-based positioning approach is that if nodes see the same redirection behavior over time, then those two nodes are likely to be near one another in the network sense. Thus, CRP

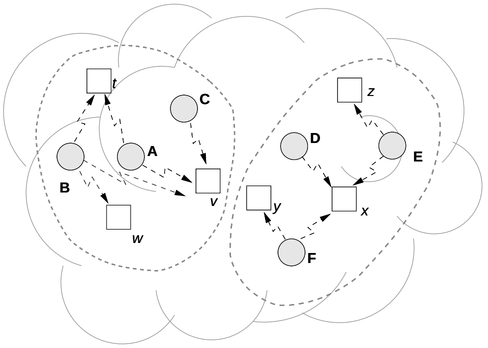

Fig. 3: CRP-based clustering. The objective is to divide a set of peers into disjoint groups (i.e., clusters) according to cluster diameter or the average distance (in terms of RTT) from the center of the cluster.

assigns two nodes to the same cluster if the cosine similarity of their ratio maps is sufficiently high. Similarly, CRP determines that two nodes are likely to be relatively far from one another if their corresponding cosine similarity is near zero.

V. Evaluation

In this section, we present experimental results showing CRP's performance in closest-node selection and clustering. It is important to note that the goal of CRP is to provide scalable relative network positions by avoiding *any* path probing and relying on pre-existing infrastructure. Thus, our evaluation is intended to show that such an approach is feasible and its performance *comparable* to more precise and correspondingly, more more costly, alternatives. We discuss the limitations and possible costs of CRP in the context of our discussion (Section VI).

To demonstrate the potential for the CRP approach, we use a set of consistently active PlanetLab nodes and a collection of DNS servers selected from the King data set [15]. We filtered the original set of DNS servers to include only those responding to ICMP pings and currently supporting recursive queries. To drive CRP, we employ the Akamai CDN [1] and gather CDN redirections using Yahoo[1] and Fox News[2]. As reported in [42], lookups to these names result in redirections that reflect dynamic network conditions. During each evaluation period, we determine CDN-based relative positions by issuing recursive DNS queries to reveal the mapping of hosts to replica servers. In parallel, we used the King measurement technique [15] to estimate "ground-truth" round-trip times (RTTs) between all pairs of hosts in the experiment. We use these RTTs to evaluate the effectiveness of CRP-based clustering. Additionally, for closest-node selection, we compare CRP performance to that of a Meridian-based service.[3] This service enjoys a large PlanetLab deployment including, at the time of our experiments, over 413 PlanetLab nodes of which 240 were consistently active. In addition, Meridian's direct-measurement approach has been shown to fare well in terms of accuracy when compared with well-known systems such as Vivaldi [11] and GNP [30]. The data presented in this section was collected between November 12-25, 2006 and January 17-23, 2007.

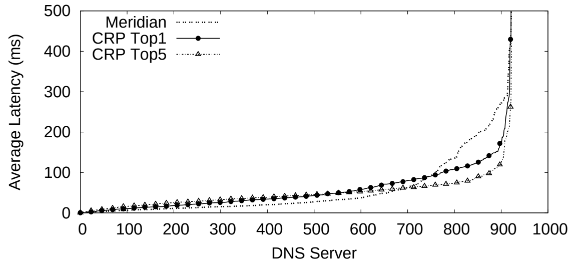

Fig. 4: CRP's closest node selection yields accuracy comparable, in terms of latency, to Meridian's.

A. Closest Node Selection

We first evaluate CRP-based closest node selection and compare responses with those provided by the Meridian-based service, For Meridian, we use the measuring PlanetLab node as the entry point to the Meridian overlay.

For this experiment, we selected 1,000 DNS servers from the King [15] data set as participants in our service. We filtered the original set to include only those servers responding to ICMP pings and currently supporting recursive queries, leaving us with a total of 4,000 hosts from which we randomly selected our 1,000 DNS servers. The 1,000 DNS servers represent *clients* requesting the identity of their closest server. We use the 240 active PlanetLab nodes running Meridian as candidate *servers* for selection. To derive a baseline for comparison, we directly measured the RTT between these PlanetLab nodes (our candidate servers) and the 1,000 different DNS servers (acting as clients in our evaluation). Finally, we used Meridian and our CRP-based approach to select the closest server (from among the PlanetLab candidates) for every client, and compared their recommendations based on the complete, RTT-based ordering of servers.

While CRP aims at supporting node selection based on relative network proximity, rather than highly accurate latency estimates, it is useful to compare the different recommendations in terms of average RTT between the client and the selected servers. Figure 4 shows this comparison for Meridian and CRP. For the latter, we plot the Top 1 and the average rank of the Top 5 recommendations. As it can be seen from the figure, about 65% of the time CRP Top 5 recommendation differs from Meridian by less than 7 ms (or about 12%) and improves over it in over 25% of the responses. For example, in about 10% of the cases, the RTT to the Meridian recommended server is more than twice that to the CRP Top 5 selected node.

The right-hand side of this graph is particularly interesting; here the latency of both recommendations differ significantly from the optimal selection based on direct measurement. To determine the unique causes of poor performance for the two approaches, we removed servers that caused poor results (relative RTTs larger than 80 ms) for both Meridian and CRP. We found that less than 20% of the servers appeared in both datasets.

[1] The Yahoo image server *us.i1.yimg.com*

[2] www.foxnews.com

[3] http://www.closestnode.com

After removing errors common to both approaches, we investigated their root causes. We found that Meridian errors can be mostly attributed to the instability of the overlay network, its broad coverage, and the known connectivity problems with some PlanetLab nodes. During its bootstrapping phase, a newly joined Meridian node will simply recommend itself as the "closest" node, effectively ignoring the request parameters. For example, the node `planetlab1.cis.upenn.edu` restarted soon after November 13, 2006, took 10 hours before responding to any requests and then, for the following 7 hours, provided itself as the closest node to all our requests. Similarly, nodes `sjtu1.6planetlab.edu.cn`, `csplanetlab3.kaist.ac.kr` and `planetlab2.eee.hku.hk` never successfully joined the Meridian overlay during our 5-day experiment, while hosts in the pairs `planetlab[1,2].iii.u-tokyo.ac.jp` and `planetlab[1,2].atcorp.com` only connected to the other host in their site. These hosts either return themselves or their collocated nodes as their closest-node recommendation to all of our requests. There were also some hops through the Meridian overlay that we have been unable to explain (e.g., going from `planetlab-01.bu.edu` in Boston, MA to the DNS server `ns1.uskonet.com` in South Africa via `plnode02.cs.mu.oz.au` in Australia).

Because CRP leverages the network view gathered by CDN infrastructures, its accuracy naturally depends on the CDN's coverage in the area of interest. CRP's poor performance in the tails of the above graphs corresponds to clients located in regions of the world that are currently poorly served by the Akamai CDN. As an example, a DNS server in New Zealand (`ns1.iconz.co.nz`) that appears in the tail of the CRP curves is redirected to 27 different replica servers including ones in Massachusetts, Tennessee and Japan.

The remaining cases for poor performance are common to both CRP and Meridian. This is simply due to the limited coverage of the selected PlanetLab nodes used in our experiments, such that some of the client nodes (i.e., DNS servers) are not near any of the used PlanetLab servers (e.g., the DNS servers in Iceland, `chopin.samskip.is` and Russia, `ns.spb.ru`).

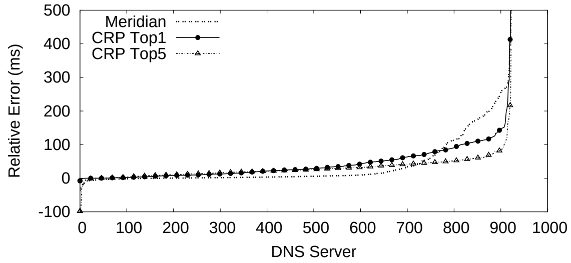

Fig. 5: Relative errors for CRP and Meridian. CRP errors do not lead to significant relative RTT differences.

Figure 5 shows the relative errors for for CRP Top 1, Top 5 and Meridian. In the case of CRP, we compute the difference between the average of the RTTs to the Top 5 recommended servers, and subtract this from the RTT to the closest node. For Meridian, we plot the difference between the RTT to the recommended server and the measured closest host. The small fraction of negative values are the result of network dynamics throughout the experiment. The figure clearly shows that most of these errors do not lead to significant relative RTT differences.

The above results for server selection illustrate the effectiveness of the CRP-based approach to relative network position. Note that, as previously stated, although our results show that both CRP and Meridian incur comparable errors, we are not advocating a CRP-based service as a replacement for more general positioning systems.

B. Clustering

We now present an evaluation of CRP-based clustering. We begin by describing our clustering technique, then provide an evaluation using a data set that contains 177 broadly distributed DNS servers as candidate nodes for clustering. For this evaluation, we assigned a set of DNS servers to each PlanetLab server, then issued recursive DNS lookups to determine CRP positions as previously described. Finally, we estimated the "ground-truth" distances among servers by using King to measure RTTs between each DNS server and the other $N-1$ DNS servers.

To place our results in the context of another low-cost technique, we compare the results of CRP-based clustering with that of ASN-based clustering. ASN-based clustering relies on the hypothesis that nodes located in the same autonomous system are nearby in a networking sense.We determine the membership of nodes to ASes according to AS numbers (ASNs) by using data from the RouteViews project; any node belonging to the same ASN is grouped into the same cluster. We acknowledge that ASN-based clustering is not optimal; however, because ASNs explicitly encode information about network structure, we believe they provide a meaningful baseline for evaluating the effectiveness of our approach.

There is a number of clustering algorithms in the literature, many of which we found inappropriate for CRP. For example, k-means and fuzzy c-means clustering require, as input, the number of clusters to form – knowledge that is not generally available to our system. Other clustering algorithms, (e.g., hierarchical) assume a node-distribution model that is unsuitable for CRP clustering of Internet hosts.

In our approach, the input to the algorithm is the set of all nodes, N, their mapping to replica server clusters and a minimum cosine similarity threshold, t. We compared various approaches to selecting initial cluster centers (e.g., random or structured based on redirection information) and assigning unclustered nodes to clusters. Ultimately, we found that a hybrid approach that we call *Strongest Mappings First (SMF)*, works best.

In this algorithm, we initially define the cluster centers as those with the strongest mappings to replica servers. Once the cluster centers have been set, the algorithm picks an unclustered node and finds its cosine similarity to each cluster center. The node is assigned to the cluster whose center produces the largest cosine similarity, if that value is greater than a threshold t. Otherwise, the node is assigned to its own cluster.

This algorithm can result in a significant number of clusters of size one, i.e., unclustered nodes. Thus, in an optional second pass of the algorithm, we select unclustered nodes at random to be cluster centers and determine if any of the other unclustered nodes belong to the cluster based on the cosine- similarity metric.

While we do not claim that SMF is the optimal CRP-based clustering algorithm, it is a simple, easily deployable approach that serves to demonstrate the feasibility of CRP-based clustering.

We begin our evaluation by examining high-level characteristics of clusters produced by our technique and by ASN-based clustering. Table I provides a summary of the results. As described above, our CRP clustering algorithm assigns a node to a cluster only if the cosine similarity between the node and a cluster center is greater than t. The first three rows in Table I demonstrate how t impacts the clustering algorithm.

A lower value for t assigns a greater fraction of nodes to clusters and also leads to a larger average cluster size. This occurs because nodes with relatively weak similarity in CDN redirection behavior are grouped into the same cluster. With a relatively large t, we see that the average cluster size is significantly lower, but the fraction of nodes assigned to clusters is also significantly smaller. The reason is that a larger t provides clusters containing only nodes with nearly identical redirection behavior and thus excludes many nodes that may have significantly similar behavior. Based on these observations, we elected to use a value for t that straddles these two extremes. Thus, for the remainder of this section we use $t = 0.1$ for CRP clustering. We chose this value because it leads to satisfactory results; however, determination of the "optimal" threshold is left as future work.

Next, we compare CRP clustering characteristics to those of ASN-based clustering. It is immediately clear that CRP finds clusters for a larger fraction of nodes (over 300% more) than ASN-based clustering. Correspondingly, it also finds more than twice the number of total clusters. This is because CRP can cluster together nearby nodes that are located in different ASes, a case that is quite common when considering a large number of nodes.

We now determine the quality of CRP clustering. A useful metric for gauging the quality of clustering in this context is to compare the average intracluster distance for nodes in a cluster to their average intercluster distance, i.e., the average distance from the center of a cluster to the center of all other clusters. If the average intercluster distance is high relative to an intracluster distance, then we are reasonably certain that our algorithm has found a good cluster.

For the remainder of this section, we limit our results to clusters with diameters smaller than 75 ms. Larger clusters are few in number and are unlikely to be useful to applications.

Figure 6 presents a CDF of *intracluster* distances for clusters formed by CRP-based clustering. The solid curve represents intracluster distances and the circular points represent the *corresponding* intercluster distances. The clustering algorithm is most effective if most of the circular points fall to the bottom right of the solid curve (the shaded region in the figure), indicating that the clustered nodes are closer to their cluster center than to the center of any other cluster. The graph shows that most of the clusters exhibit a diameter of less than 40 ms, and members of those clusters are in fact far from other cluster centers. Thus, CRP provides the ability to form high-quality node clusters without directly probing each node.

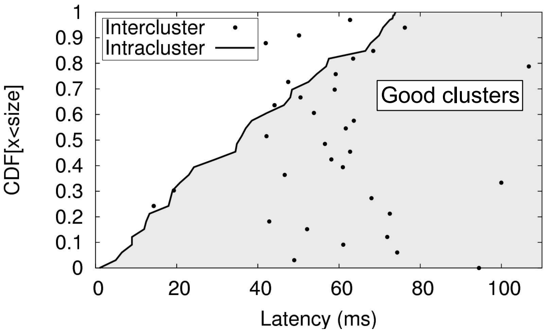

Fig. 6: CDF of intra- and inter-cluster distances. Good clusters are located in the shaded region.

Finally, we evaluate whether the quality of clusters generated by CRP is on par with those generated by ASN-based clustering. Intuitively, ASes should provide high-quality clusters when the AS does not span multiple geographic regions, a case that is common in our dataset. As we noted above, CRP produces a larger number of clusters and includes a larger portion of candidate nodes; in this evaluation we determine whether CRP results on lower-quality clusters as a result of this broad coverage.

For this analysis, we group clusters into buckets of diameter 0–25 ms and 25-75 ms, then count the number of "good" clusters (i.e., those in the shaded region in Fig. 6) in each bucket for each algorithm. Figure 7 demonstrates that CRP clustering finds over 50% more high-quailty clusters in the first bucket and more than double the number of clusters in the second bucket. As already pointed out, this is due to CRP ability to find clusters with nodes in multiple ASes.

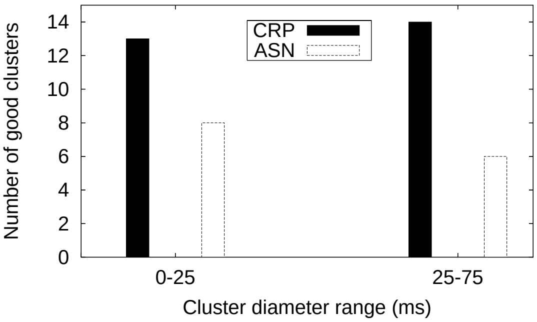

Fig. 7: Number of good clusters with intracluster distance within each bucket range.

VI. Discussion

The designer of a relative network positioning service is faced with clear tradeoffs between accuracy and measure-

Technique	# nodes clustered	% nodes clustered	# of clusters	[mean, median, max] cluster size
CRP (t=0.01)	131	74%	35	[3.74, 3, 21]
CRP (t=0.1)	**128**	**72%**	**36**	**[3.56, 3, 12]**
CRP (t=0.5)	114	64%	38	[3.00, 2, 9]
ASN	41	23%	16	[2.56, 2, 5]

TABLE I: Summary statistics for clusters formed by CRP (using various cluster-membership thresholds, t) and ASN-based clustering.

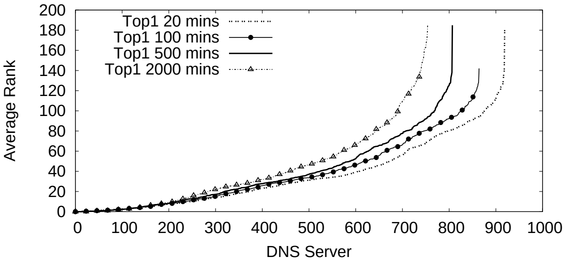

Fig. 8: Average rank for different probe frequencies (lower rank is better). An effective service can be based on a request interval as low as 100 minutes – a virtually insignificant overhead.

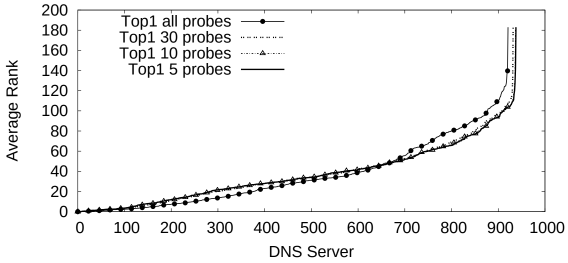

Fig. 9: Average rank for different probe window sizes, i.e., the number of observations needed to make a useful estimation. A small, 10-probe, window size is sufficient for effective CRP-based server selection.

ment overhead, deployment cost, ease of use and turnaround time [34]. For example, although increasing the number of direct measurements will certainly improve the accuracy of an estimate [46], [47], it may also render the service non-scalable. Similarly, while estimate accuracy could be improved via longer turnaround times [46], many applications require quick responses to make their decisions.

CRP offers a new point in this space – a lightweight and highly scalable approach to perform node selection based on relative network positioning with accuracy comparable to that of alternative approaches. CRP's achieves these properties by avoiding additional direct measurements to either landmarks or other peers in an overlay. Instead, it strategically reuses the network views gathered by large-scale CDNs. In addition, CRP is easy to deploy and use, as it leverages existing CDN infrastructures without imposing an unduly large load on them, and provides a well-known interface that simplifies application integration.

CRP can form the basis of service that is commensalistic with symbiotic CDNs, where client applications benefit by reusing CDNs' network monitoring information without negatively impacting CDN services. Early results indicate that is indeed possible. Based on our closest node selection experiment (Sec. V), we explored the potential of different intervals between redirection requests and the effect of window sizes on CRP's relative positioning estimations.

Figure 8 shows the average ranks resulting from request intervals of 20 min, 100 min, 500 min and 2000 min. Rank is set to the index of the recommended server in the RTT-based ordered list. For example, if the node selected by a given approach is the first one in the list, the result is assigned a rank of 0. If the recommended "closest" node is the fifth one in the list, the rank value of the result is 4.

Note that, as a side effect of extending the probe interval, some DNS servers may not be able to find PlanetLab nodes with common replica servers during the duration of our experiments. This explains the smaller number of DNS servers for which average rank is plotted. Clearly, an effective service can be based on request intervals as low as 100 minutes. When considering that the Akamai CDN sets its DNS entry's TTL to 20 seconds, a probing interval of almost two hours means that a CRP client will generate an additional load significantly lower than what is expected from an ordinary web client. Furthermore, even this minor overhead may not be necessary if the service can passively monitor user-generated DNS translations (e.g., from Web browsing) instead of actively requesting CDN redirections.

Once a probing interval is set, a related question is the number of redirections necessary to make a useful estimation of relative network positioning. This would help determine the overhead of a CRP-based service and the bootstrapping time of a CRP node, i.e., the time before an effective CRP-based decision can be made from the first observed redirection. Again based on the closest node selection experiment, we illustrate the potential impact of probe window sizes, i.e., the number of recent redirections considered in a recommendation. Figure 9 shows this with a probe interval fixed at 10 minutes. There are a few clear points to draw from the figure. While a 30-probe window size offers some small improvement, a window size of 10 probes seem to be sufficient for effective CRP-based closest node selection. Also, given a 10-probe window size and a probe interval of 10 minutes, a CRP client will need a bootstrapping time of ∼100 minutes. Finally, as it can be seen from the graph, the "all probes" curve shows better average rank for two-thirds of the DNS servers than what is possible with more limited window sizes. However, for the remainder of the DNS servers, using all probes yields worse average rank than looking at the last 10-30 probes. This is primarily due to variable network dynamics: in more stable environments, maintaining longer histories helps to refine our results, while longer histories in an environment with more dynamic conditions can actually harm overall performance by incorporating stale information.

In this paper, we hand-picked the CDN names to use based on historical empirical data; however, in practice, it is preferable to use an approach that selects CDN names based on the quality of relative position information that they provide. One way to do this is to ping the replica servers returned for each CDN name during the bootstrapping phase and use only those names corresponding to low-latency servers. While this approach requires a small amount of active probing, the overhead is a small and independent of the number of nodes in the system. If one requires an adaptive solution that does not perform any active probing, one can eliminate those CDN names that return replica servers that do not provide positioning information. For example, our experiments have shown that when the Akamai CDN returns replica servers with IP addresses owned by the Akamai domain, those servers are often far away from the node performing the DNS lookup. Based on this observation, one can use simple filtering rules to select only CDN names that do not return such servers.

VII. Conclusions

In this paper, we introduced a new approach to relative network positioning that avoids direct probing by leveraging the dynamic association of nodes with replica servers from a large content distribution network. We call this approach *CRP* for *CDN-based Relative network Positioning.* We described a CRP-based positioning service that is lightweight, highly scalable and easy to deploy and use. We apply CRP to two common location problems in distributed systems: closest node selection and clustering. Results from a wide-area evaluation with over 1,200 hosts show that CRP offers accuracy comparable to that of alternative approaches with virtually no overhead on the CDNs it relies on, thus indicating that the technique can be implemented as a commensalistic service with CDNs. An open problem that directly follows from this work is to understand how a CRP-based service can be combined with previously proposed latency-prediction approaches into a service that offers relative network positioning between arbitrary hosts with little-to-no overhead.

VIII. Acknowledgements

We would like to thank the anonymous ICDCS reviewers for their valuable feedback on earlier versions of this paper. This research was partially funded by the National Science Foundation under CAREER Award CNS-0644062. Opinions and findings expressed in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NSF or the U.S. government.

References

[1] Akamai. Akamai CDN. http://www.akamai.com.

[2] Banerjee, S., Bhattacharjee, B., and Kommareddy, C. Scalable application layer multicast. In *Proc. of ACM SIGCOMM* (August 2002).

[3] Bharambe, A., Pang, J., and Seshan, S. Colyseus: A distributed architecture for online multiplayer games. In *Proc. of the USENIX Symposium on Networked System Design and Implementation (NSDI)* (May 2006).

[4] Bornstein, C., Canfield, T., and Miller, G. Overlay routing networks (Akarouting). http://www-math.mit.edu/steng/18.996/lecture9.ps, April 2002.

[5] Castro, M., Druschel, P., Kermarrec, A., and Rowstron, A. Scribe: A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in Communication 20*, 8 (October 2002).

[6] Chen, Y., Lim, K. H., Katz, R. H., and Overton, C. On the stability of network distance estimation. *SIGMETRICS Performance Evaluation Review 30*, 2 (September 2002).

[7] Chu, Y.-H., Rao, S. G., Seshan, S., and Zhang, H. A case for end system multicast. *IEEE Journal on Selected Areas in Communication 20*, 8 (October 2002), 1456–1471.

[8] Clark, D. D., Partridge, C., Ramming, J. C., and Wroclawski, J. T. A knowledge plane for the Internet. In *Proc. of ACM SIGCOMM* (August 2003).

[9] Cohen, B. Incentives build robustness in BitTorrent. In *Proc. of the Workshop on Economics of Peer-to-Peer Systems (P2PEcon)* (June 2003).

[10] Costa, M., Castro, M., Rowstron, A., and Key, P. PIC: Practical Internet coordiantes for distance estimation. In *Proc. of the IEEE International Conference on Distributed Computing Systems* (March 2004).

[11] Dabek, Cox, Kaashoek, and Morris, R. Vivaldi: A decentralized network coordinate system. In *Proc. of ACM SIGCOMM* (August-September 2004).

[12] Francis, P., Jamin, S., Jin, C., Jin, Y., Raz, D., Shavitt, Y., and Zhang, L. IDMaps: A global Internet host distance estimation service. *IEEE/ACM Transactions on Networking 9*, 5 (October 2001).

[13] Gadde, S., Chase, J., and Rabinovich, M. Web caching and content distribution: a view from the interior. In *Proc. of the Workshop on Web Content Caching and Distribution (WCW)* (May 2000).

[14] Gibbons, P., Karp, B., Ke, Y., Nath, S., and Seshan, S. IrisNet: an architecture for a world-wide sensor web. *IEEE Pervasive Computing 2*, 4 (2003).

[15] Gummadi, K. P., Saroiu, S., and Gribble, S. D. King: Estimating latency between arbitrary Internet end hosts. In *Proc. ACM Internet Measurement Workshop* (November 2002).

[16] Johnson, K., Carr, J., Day, M., and Kaashoek, M. The measured performance of content distribution networks. In *Proc. of the Workshop on Web Content Caching and Distribution (WCW)* (May 2000).

[17] Kangasharju, J., Ross, K., and Roberts, J. Performance evaluation of redirection schemes in content distribution networks. *Computer Communications 24*, 2 (2001), 207–214.

[18] Koletsou, M., and Voelker, G. The Medusa proxy: A tool for exploring user-perceived web performance. In *Proc. of the Workshop on Web Content Caching and Distribution (WCW)* (June 2001).

[19] Krishnamurthy, B., and Wills, C. Analyzing factors that influence end-to-end web performance. In *Proc. of the Workshop on Web Content Caching and Distribution (WCW)* (April 2000).

[20] Krishnamurthy, B., Wills, C., and Zhang, Y. On the use and performance of content distribution networks. In *Proc. ACM Internet Measurement Workshop* (Nov. 2001).

[21] Ledlie, J., Gardner, P., and Seltzer, M. Network coordinates in the wild. In *Proc. of USENIX NSDI* (April 2007).

[22] Lee, K.-W., Ko, B.-J., and Calo, S. Adaptive server selection for large scale interactive online games.

[23] Lehman, L., and Lerman, S. PCord: network position estimation using peer-to-peer measurements. In *In Proc. of the Symposium on Network Computing and Applications* (August 2004).

[24] Lim, H., Hou, J., and Choi, C. Constructing Internet coordinate system based on delay measurement. In *Proc. of the Internet Measurement Conference* (Oct. 2003).

[25] LimeLight Networks. Limelight networks CDN. http://www.limelightnetworks.com.

[26] Madhyastha, H. V., Anderson, T., Krishnamurthy, A., Spring, N., and Venkataramani, A. A structural approach to latency prediction. In *Proc. of IMC* (October 2006).

[27] Madhyastha, H. V., Isdal, T., michael Piatek, Dixon, C., Anderson, T., Kirshnamurthy, A., and Venkataramani, A. iPlane: an information plane for distributed systems. In *Proc. of USENIX OSDI* (November 2006).

[28] Mirror Image. Mirror image CDN. http://www.mirror-image.net.

[29] Nakao, A., Peterson, L., and Bavier, A. A routing underlay for overlay networks. In *Proc. of ACM SIGCOMM* (August 2003).

[30] Ng, T., and Zhang, H. Predicting Internet network distace with coordinates-based approaches. In *Proc. of IEEE INFOCOM* (June 2002).

[31] Ng, T., and Zhang, H. A network positioning system for the Internet. In *Proc. of the USENIX Annual Technical Conference* (June 2004).

[32] Padmanabhan, V. N., Wang, H. J., and Chou, P. A. Resilient peer-to-peer streaming. In *Proc. of IEEE ICNP* (November 2003).

[33] Pias, M., Crowcroft, J., Wilbur, S., Harris, T., and Bhatti, S. Lighthouses for scalable distributed locations. In *Proc. of the International Workshop on Peer-to-Peer Systems (IPTPS)* (Feb. 2003).

[34] Pietzuch, P., Ledlie, J., and Seltzer, M. Supporting network coordinates on PlanetLab. In *Proc. of the Workshop on Real, Large Distributed Systems (WORLDS)* (December 2005).

[35] R, H., Hellerstein, J., Boona, N., Loo, T., Shenker, S., and Stoica, I. Querying the Internet with PIER. In *Proc. of VLDB* (September 2003).

[36] Ratnasamy, S., Handley, M., Karp, R., and Shenker, S. Topologically-aware overlay construction and server selection. In *Proc. of IEEE INFOCOM* (June 2002).

[37] Rowstron, A., and Druschel, P. Storage management and caching in PAST, a large-scale, persistant peer-to-peer storage utility. In *Proc. of the ACM Symposium on Operating System Principles* (October 2001).

[38] Salton, G., and McGill, M. J. *Introduction to modern information retrieval*. McGraw-Hill, New York, NY, 1986.

[39] Saroiu, S., Gummadi, K., Dunn, R., Gribble, S., and Levy, H. An analysis of Internet content delivery systems. In *Proc. of the USENIX Operating Systems Design and Implementation (OSDI)* (December 2002).

[40] Shaikh, A., Tewari, R., and Agrawal, M. On the effectiveness of DNS-based server selection. In *Proc. of IEEE INFOCOM* (Anchorage, AK, April 2001).

[41] Shavitt, Y., and Tankel, T. Big-bang simulation for embedding network distances in euclidean space. In *Proc. of IEEE INFOCOM* (Apr. 2003).

[42] Su, A.-J., Choffnes, D. R., Kuzmanovic, A., and Bustamante, F. E. Drafting behind Akamai: Travelocity-based detouring. In *Proc. of ACM SIGCOMM* (September 2006).

[43] Tang, L., and Crovella, M. Virtual landmarks for the Internet. In *Proc. of the Internet Measurement Conference* (October 2003).

[44] Wang, L., Park, K., Pang, R., Pai, V., and Peterson, L. Reliability and security in the CoDeeN content distribution network. In *Proc. of the USENIX Annual Technical Conference* (June 2004).

[45] Wawrzoniak, M., Peterson, L., and Roscoe, T. Sophia: An information plane for networked systems. In *Proc. of HotNets* (November 2003).

[46] Wong, B., Slivkins, A., and Sirer, E. Meridian: A lightweight network location service without virtual coordinates. In *Proc. of ACM SIGCOMM* (Apr. 2005).

[47] Zhang, R., Tang, C., Hu, Y. C., Fahmy, S., and Lin, X. Impact of the inaccuracy of distance prediction algorithms on Internet applications - an analytical and comparative study. In *Proc. of IEEE INFOCOM* (April 2006).