Identifying and Characterizing Anycast in the Domain Name System

USC/ISI Technical Report ISI-TR-671, May 2011*

Xun Fan
Information Sciences Institute
Univ. of Southern California
xunfan@isi.edu

John Heidemann
Information Sciences Institute
Univ. of Southern California
johnh@isi.edu

Ramesh Govindan
Computer Science Dept.
Univ. of Southern California
ramesh@usc.edu

ABSTRACT

Since its first appearance, IP anycast has become essential for critical network services such as the Domain Name System (DNS). Despite this, there has been little attention to independently identifying and characterizing anycast nodes. External evaluation of anycast allows both third-party auditing of its benefits, and is essential to discovering benign masquerading or hostile hijacking of anycast services. In this paper, we develop ACE, an approach to identify and characterize anycast nodes. ACE first method is DNS queries for CHAOS records, the recommended debugging service for anycast, suitable for cooperative anycast services. Its second method uses *traceroute* to identify all anycast services by their connectivity to the Internet. Each individual method has ambiguities in some circumstances; we show a combined method improves on both. We validate ACE against two widely used anycast DNS services that provide ground truth. ACE has good precision, with 88% of its results corresponding to unique anycast nodes of the F-root DNS service. Its recall is affected by the number and diversity of vantage points. We use ACE for an initial study of how anycast is used for top-level domain servers. We find one case where a third-party server operates on root-DNS IP address, masquerades to capture traffic for its organization. We also study the 1164 nameserver IP addresses that cover all generic and country-code top-level domains, gather evidence that at least 14% and perhaps 32% use anycast.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network topology*; C.2.5 [**Computer-Communication Networks**]: Local and Wide-Area Networks—*Internet*; C.2.6 [**Computer-Communication Networks**]: Internetworking

General Terms

Measurement

Keywords

IP Anycast, Root name severs, TLD name servers

1. INTRODUCTION

To provide rapid response and high availability to customers, many large network services are distributed globally. For some of these services, the abstraction of a single logical service is provided by servers in multiple locations with similar or identical replicated contents accessed using a single logical identifier. Content delivery networks (for example, [12]), mirroring services (for example, [11]), URNs [33], and IP anycast [28] all fit this model.

IP anycast is one mechanism to provide an abstraction for a logical service distributed across the Internet. As defined in RFC-1546 and RFC-4786, an anycast service operates on a single IP address, but multiple *anycast nodes* replicate that service at different physical locations. Each node may be implemented with one or more *servers* (a physical or virtual machine), each of which listens to both the anycast address and one or more unicast addresses as well. Standard interdomain routing directs clients to the nearest replica and handles fail-over to other nodes as required. (We review IP anycast details and terms in Section 2.1.)

Anycast is used for many core services of the Internet today. It is widely used for DNS [15]: as of April 2011, 10 out of 13 root name servers employ anycast [30], and in Section 4.2 we show that anycast is used in many other top-level domain servers. It is also used for discovering IPv6-to-IPv4 relay routers [17], load distribution [36, 13], sinkholes [14], and similar services. Anycasted services benefit from anycast's load balancing and ability to mitigate denial-of-service attacks [1], and research proposals have discussed improvements to scale to many anycast destinations [19].

The use of anycast for core Internet services motivates

*This work is partially supported by the United States Department of Homeland Security contract numbers N66001-10-C-0081 ("AMITE") and NBCHC080035 ("LANDER-2007"). All conclusions of this work are those of the authors and do not necessarily reflect the views of DHS.

the need for tools to understand anycast use. Since DNS is central to the performance of web services (see for example [34]) and content delivery networks, these tools and methods can help Internet service operators understand how anycast affects their overall performance. Although anycast providers can directly monitor the status of their service, clients of anycast operators can use these tools to audit the service they are purchasing, and providers can obtain a "client's-eye" view from external observations of their anycast service. By serving the same IP address from multiple locations, anycast uses the same mechanisms as route prefix hijacking, but for good. Yet anycast services themselves can be subject to partial hijacking, as we observe in Section 4.1. Finally, while anycast today is used mainly for DNS, 6to4 traffic, load distribution, and sinkholes, an understanding of the current deployment may spur additional deployments. Although anycast performance has been studied by a number of groups (we cover related work in Section 5), these needs motivate our study of anycast *discovery* and *mapping*.

The first contribution of our work therefore is to propose ACE (Anycast Characterization and Evaluation), a set of new methods to identify anycast nodes in Section 2. We begin with specific DNS query targets that are the debugging methods anycast operators have built and often employ in their systems [38]. This method requires a cooperative and correctly configured target. We therefore also develop a traceroute-based method that only depends on anycast's inherent characteristic: traffic to the same IP address follows different paths. We show that these methods complement each other with a combined algorithm that reduces observation error.

The second contribution of our work is to validate ACE and evaluate what factors affect its performance (Section 3). We apply our methods from 242 PlanetLab nodes, identifying the anycast infrastructure for the F-root domain servers, and for Packet Clearing House's DNS servers, both of which provide ground truth. We look at the coverage of our two data sources and the accuracy of the two methods and their combination. The precision of our results, how many answers are correct anycast nodes, varies from 58% to 100% depending on information source and target. Recall, how many of the ground truth anycast nodes we find, varies from 38% to 49%. Recall is dominated by the number and diversity of vantage points, so we study this factor explicitly.

Our final contribution is to apply ACE to understand how anycast is used in practice over many domains. We report two findings. First, in the process of validation, we found a third-party DNS server masquerading as an anycast node for a public root server (Section 4.1). Although known to the root operator, ACE demonstrates the importance of independent techniques to audit critical Internet infrastructure. Second, in Section 4.2, we apply ACE to servers for all generic and country-code top-level domains (gTLDs and ccTLDs). Although we cannot judge the size of each anycast instance (because of the uncertain recall of our algorithms given our current vantage points), our data suggests hundreds of active anycast services in use, and gives some information about how many unique anycast providers are in operation. Although our findings are not definitive, since our observation methods are imperfect, to our knowledge they are the first study of anycast use (as opposed to performance) for TLD name services.

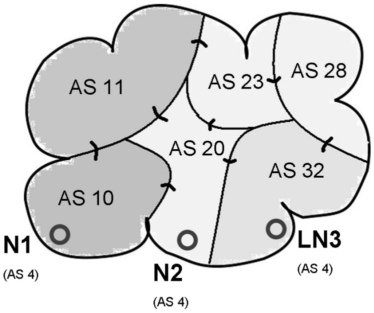


Figure 1: Anycast and routing: three anycast nodes (N1, N2, and LN3) and their cacthments (dark, light and medium grey regions).

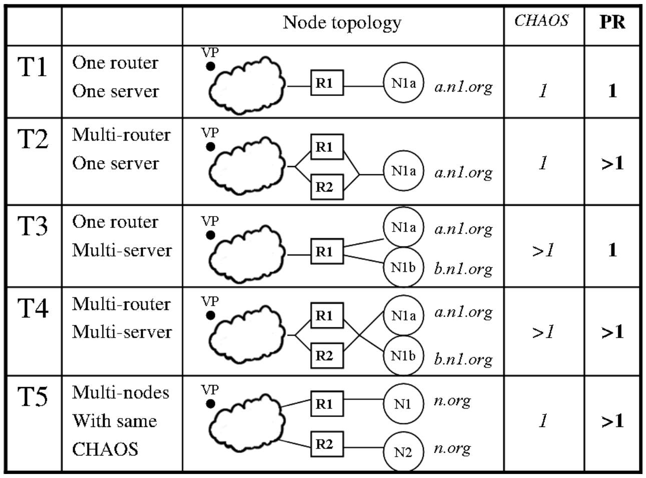
2. METHODS FOR ANYCAST DISCOVERY

We now describe our three methods to characterize anycast services. Our DNS-query-based method uses operator-configured information, our traceroute-based method uses only network paths, and our hybrid algorithm combines both. Before defining the algorithms, we review anycast terminology.

2.1 Anycast Background

IP anycast provides clients of a distributed service with a single-server abstraction [28]. Clients of the service send traffic to a designated IP address identifying the *anycast service*. However, the service itself is implemented by a service provider using one or more *anycast nodes* that can be physically distributed around the Internet. Standard routing protocols ensure that the user's packets are sent to a nearby anycast node. Because routing, and therefore node selection, is independent of communication, anycast is usually used only for stateless, single-packet-exchange services like DNS [15] or datagram relay [17].

Figure 1 shows how three anycast nodes might cover a network of six ASes. Looking inside each node, Figure 2 shows the complexity that is possible in practice as anycast nodes are implemented by single or multiple machines at a given site, each with one or more network



		Node topology	*CHAOS*	**PR**
T1	One router One server	VP R1 N1a *a.n1.org*	*1*	**1**
T2	Multi-router One server	VP R1 R2 N1a *a.n1.org*	*1*	**>1**
T3	One router Multi-server	VP R1 N1a *a.n1.org* N1b *b.n1.org*	*>1*	**1**
T4	Multi-router Multi-server	VP R1 R2 N1a *a.n1.org* N1b *b.n1.org*	*>1*	**>1**
T5	Multi-nodes With same CHAOS	VP R1 R2 N1 *n.org* N2 *n.org*	*1*	**>1**

Figure 2: Anycast node configurations, and CHAOS-related observations in italics and traceroute penultimate routers in bold.

interfaces. In addition, different routing options can be used control anycast scope. We review these options next, using the terminology defined in RFC-4786 [1].

Routing options.

The area covered by a given anycast node is its *catchment.* Anycast nodes have two levels of external visibility: global and local. In Figure 1, anycast nodes N1 and N2 are global, each with catchments encompassing multiple ASes (AS11 and AS12; and ASes 21, 22, and 23, respectively),

Node LN3 is local and so affects only AS31. Global nodes advertise an anycast IP prefix to BGP and are visible Internet-wide [1]. Local nodes advertise an anycast IP prefix with the *no-export* BGP attribute [2] and are visible only to adjacent autonomous systems. Larger anycast services often include both local and global nodes, but either may be omitted.

Anycast is available in both IPv4 and IPv6. We examine only its use for IPv4.

Options inside each anycast node.

While routing allows clients to access a nearby anycast node, there can be complexity inside the anycast node as well, as one or more servers may provide the service. Figure 2 enumerates the key features of all important configurations that we have encountered.

The top row (T1) shows the simplest case, where a single server provides service at a given anycast node. That anycast node listens to traffic on both the anycast address that is shared with all other anycast nodes, but also on a second, unique unicast address used for management.

Since anycast nodes are often placed in Internet exchanges (IXP) with complex local topologies, row T2 shows a single machine with multiple adjacent routers, connected by a shared LAN, or a sever with multiple network interfaces.

For large services such as a top-level domain server, a service at an anycast node may be provided by multiple physical servers. Cases T3, T4, and T4 show multiple servers behind one (T3) or two or more (T4) routers.

Finally, case T5 shows two different anycast nodes (N1 and N2), but with identical CHAOS records. Such use is inconsistent with their specification, but may occur due to accident or hijacking. We cannot distinguish T5 from T2 by external observation; we see neither case in our ground truth but do observe such cases in our study of TLDs (Section 4.2).

Identifying anycast nodes.

This review of anycast node configurations suggests the two basic methods we use to identify anycast nodes. Sometimes anycast servers or nodes are self-identifying using DNS CHAOS records, as shown by the italic labels next to the machines in Figure 2. This mechanism is the diagnosis method recommended by anycast operators. In Section 2.2 we describe how one can directly query the servers. The target must have correctly configured

DNS to support these queries; case T5 shows a risk with this method: two different nodes mis-configured with the same CHAOS TXT record may not be distinguishable.

As an alternative, we also examine the *penultimate router* (PR) on a traceroute to the anycast node (Section 2.3). This method depends more on details of the network configuration of the anycast node, so the distinctions between T1 and T2, or T3 and T4 can cause possible measurement error. We use this method to identify non-cooperative or misconfigured anycast nodes, and to learn the different naming schemes providers use for CHAOS records.

2.2 CHAOS DNS Queries

Operators of anycast services must have a means of observing and debugging their services. RFC-4892 defines a set of conventions using DNS records to identify individual anycast nodes and servers. Although not mandatory, we find that these conventions appear to be widely followed (Section 4.2).

Since anycast is widely used in global DNS services, and since DNS provides a convenient external query mechanism, RFC-4892 suggests how DNS can be used to identify a specific anycast server [38]. It re-purposes CHAOS network records (from the now defunct Chaosnet [25], an alternative to IP), supported in the DNS standard [24] for this purpose. Queries for the name *hostname.bind* or *id.server* with type TXT and class CHAOS are defined to return a unique string. Unfortunately, the contents of this are not formally standardized, so each provider selects some scheme to uniquely identify their servers. We decode some of these schemes in Section 4.3. We use the term *CHAOS DNS query* to refer to this mechanism in this paper.

In principle, presence of these records should make identifying anycast servers trivial. Standard DNS tools (such as dig or nslookup) can retrieve this information. Because CHAOS records are tied to individual servers, they correctly identify single-server nodes (cases T1 and T2 in Figure 2) and can also detect each server in multi-server nodes (cases T3 and T4).

However, in practice CHAOS records are not always be sufficient to identify anycast servers. CHAOS records are specified in an informational RFC, and not in a mandated standard. Thus anycast providers may choose not to provide them. They were also defined initially in the BIND DNS implementation (in fact, using a record that includes the name "bind"), and availability in other implementations has come later. In addition, CHAOS records indicate anycast servers, but conventions to relate anycast servers to nodes are unspecified. Thus the multi-server cases T3 and T4 in Figure 2 require additional information to determine the two servers located at the anycast node. We show in Section 3.3 that a standard here can avoid overcounting servers for nodes, improving precision for some cases, and in Section 4.3 we consider naming conventions in top-level domains. Third, as described above, CHAOS records may be misconfigured (case T5 of Figure 2). Finally, as we discuss later, a DNS masquerader or hijacker may intentionally omit or provide duplicate CHAOS records.

For these reasons we next describe a second mechanism to identify anycast nodes using traceroute.

2.3 Traceroute

In this section, we discuss how to use *traceroute* to track the path to the server from a vantage point. In Section 2.5 we show how traceroutes from many locations are combined to characterize the entire anycast service.

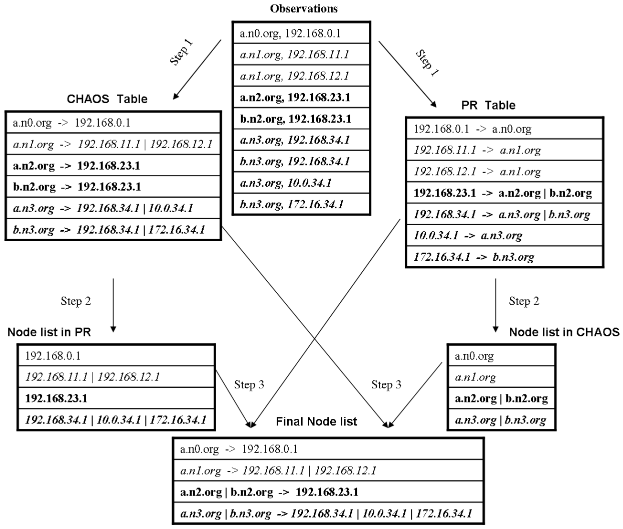
Traceroute finds part or all of the network path from the prober to a nearby anycast node. We simplify the path and focus on the *penultimate router*, or *PR*. Our hypothesis behind this method is that each anycast node will have one PR, as in a node of type T1 in Figure 2.

In practice, this hypothesis is only partially correct, since anycast nodes with a rich local topology sometimes have multiple PRs (case T2 of Figure 2) or multiple servers per node (cases T3 and T4). These cases complicate our analysis. Since these routers are nearly always co-located with the anycast node in the same IXP, we use simple heuristics to partially address this problem. We assume routers with "nearby" addresses are in the same IXP; currently, we define nearby as within the same /24 address block. When we cannot identify multiple PRs as nearby, our traceroute method overestimates the number of anycast nodes. In Section 3.4 we evaluate how often multiple PRs cause overcounts, and in the next section we show how a combination of the methods can help. Development of better PR alias resolution is an area of ongoing work.

2.4 Combining CHAOS and Traceroute

Each of our observation methods (CHAOS and Traceroute) works best when it can determine a single identifier per anycast node. As shown in Figure 2, our ability to determine identifiers depends on the anycast node and network configuration. Sometimes both methods work (case T1), or one of the two works (cases T2, T3, and T5). In case T4, both methods fail with an overcount of the anycast node, and when no vantage point is in the node's catchment, they undercount. The fact that the two approaches have complementary success cases suggests we can combine them, looking at both the PRs and CHAOS records together.

To combine the methods, we observe that if either method results in a single identifier, we know there is a single anycast node, even if the other suggests multiple



Observations
a.n0.org, 192.168.0.1
a.n1.org, 192.168.11.1
a.n1.org, 192.168.12.1
a.n2.org, 192.168.23.1
b.n2.org, 192.168.23.1
a.n3.org, 192.168.34.1
b.n3.org, 192.168.34.1
a.n3.org, 10.0.34.1
b.n3.org, 172.16.34.1

CHAOS Table
a.n0.org -> 192.168.0.1
a.n1.org -> 192.168.11.1 | 192.168.12.1
a.n2.org -> 192.168.23.1
b.n2.org -> 192.168.23.1
a.n3.org -> 192.168.34.1 | 10.0.34.1
b.n3.org -> 192.168.34.1 | 172.16.34.1

PR Table
192.168.0.1 -> a.n0.org
192.168.11.1 -> a.n1.org
192.168.12.1 -> a.n1.org
192.168.23.1 -> a.n2.org | b.n2.org
192.168.34.1 -> a.n3.org | b.n3.org
10.0.34.1 -> a.n3.org
172.16.34.1 -> b.n3.org

Node list in PR
192.168.0.1
192.168.11.1 | 192.168.12.1
192.168.23.1
192.168.34.1 | 10.0.34.1 | 172.16.34.1

Node list in CHAOS
a.n0.org
a.n1.org
a.n2.org | b.n2.org
a.n3.org | b.n3.org

Final Node list
a.n0.org -> 192.168.0.1
a.n1.org -> 192.168.11.1 | 192.168.12.1
a.n2.org | b.n2.org -> 192.168.23.1
a.n3.org | b.n3.org -> 192.168.34.1 | 10.0.34.1 | 172.16.34.1

Figure 3: An example showing combining CHAOS records and traceroute.

nodes. Figure 3 shows this process. We take observations from all vantage points as input, then separately merge records with duplicate CHAOS and PR identifiers. We then merge these lists to get our combined estimate, as shown in the example at Figure 3.

This example shows how the combined method can resolve ambiguities when observing cases T2, T3 and T4 of Figure 2. Node *n1* has the topology of case T2, node **n2** has the topology of case T3 and node ***n3*** has the topology of case T4. At the first step, the PR Table groups multiple servers to a single node, assisting for cases T3 and T4 (top right, where PR 192.168.34.1 groups servers ***a.n3.org*** and ***b.n3.org***). The CHAOS Table also groups multiple PRs in a node (top left, grouping two PRs of node *n1* and four of ***n3***). Then in second step we join these results with each other. The algorithm is symmetric; only one side or the other is required to produce the same result.

2.5 Observing from Many Vantage Points

CHAOS records and traceroute provide data about specific anycast nodes. To build a picture of the anycast service as a whole, we need to apply those methods from many different *vantage points* (VPs) spread around the Internet.

Each vantage point is in the catchment of some anycast nodes. Since standard routing allows only one best route to each IP address, our coverage of the anycast service is limited by our number of vantage points. However, Figure 4 shows some of the challenges that arise when combining traceroute observations from different vantage points. VP *a* observes node N1. VP *b* also observes N1, and with the same PR, so we detect these as equivalent observations. VP *c* observes N1, but with a different PR, suggesting that N1 has multiple adjacent routers as in type T2 of Figure 2. Anycast node N2 has two servers (Figure 2, type T4), and vantage points *d* and *e* detect each one separately. Finally, there are no vantage points in the catchment for LN3, so it goes unobserved.

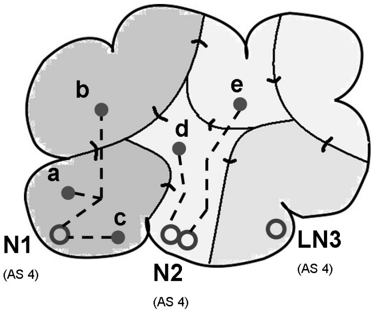


Figure 4: Observing anycast nodes N1, N2, and LN3 from vantage points *a* to *e*.

Similar challenges arise when using CHAOS records. While VPs *a*, *b*, and *c* will observe identical CHAOS records for N1, the results from VPs d and e depend on how the two servers at N2 are configured. And LN3 still is unobserved.

We evaluate the effect of number of vantage points on both methods and their combination in Section 3.6.

3. VALIDATION

We next evaluate the accuracy of each of our methods as compared to ground truth, and the effects of the number of vantage points on our results.

3.1 Validation Methodology

To validate, we run our three methods (CHAOS DNS queries, traceroute, and combined) from many global vantage points to two large anycast services for which we have ground truth.

Vantage points: We gather our data from 242 PlanetLab nodes at 238 unique sites in 40 different countries. At four sites we use two nodes to cope with node outages. Each node runs our data gathering scripts, using traceroute and dig to do the actual queries. We gather all observations at a central site for analysis.

Targets: We consider two targets: the F-root DNS server run by ISC, and the Packet Clearing House Anycast service that hosts 37 ccTLDs and 1 gTLD, and provides secondary service for 18 more. We selected these providers as targets for two reasons. First, both are large, professionally run anycast providers serving dozens of important major domains—thus, they are representative of other major anycast providers. Second, both are non-profit organizations that emphasize outreach to research and provide public descriptions of their infrastructure. Their willingness to respond to our queries about their infrastructure was essential to gaining faith in our validation.

Ground truth: We consider two types of ground truth: oracle truth, and authority truth. By *oracle truth*, we mean the actual set of nodes that respond to an anycast address in the Internet at any instant. We identify it as "oracle" truth because defining it requires a perfect snapshot of network routing from all parts of the Internet—an impossible goal. We define *authority truth* (written as A) as the list of anycast nodes that we get from the anycast service operator.

Oracle and authority truth can diverge for two reasons. First, third parties may operate anycast nodes for a given service with or without the provider's knowledge. Such third party nodes would not be part of authority truth. We discuss an example of a third-party node for F-root in Section 4.1. Second, we derive authority truth from public web pages. We find these web pages sometimes lag the current operational system, as discussed in Section 3.3.

3.2 Metrics

We evaluate our approach using precision and recall, metrics taken from the field of information retrieval. We refine these terms for our application to recognize different error cases and authority and oracle ground truth.

To review these terms and how we apply them in our case, consider Figure 4 where five vantage points (a through e) probe three anycast nodes. Probes from a through c find N1, and d and e find N2, the *true positives*. Node LN3 is omitted because there are no vantage points in its catchment, so it is a *false negative* (an undercount). To correct this error, we need a new vantage point in LN3's catchment; we study this relationship in Section 3.6.

There are three cases that might be classified as false positives. If we are unable to distinguish that two machines at N2 represent a single anycast node, then we would *overcount*. In an overcount, neither observation is completely wrong (since both N2a and N2b *are* anycast servers), but they result in a mis-estimate of the number of anycast *nodes*. When we detect a node that we confirm is operated by the anycast provider but is not in authority truth, we have an *missing authority* ("missauth" for short). Finally, if a non-authorized anycast node appeared in the AS with vantage point b, we record an *extra* node. An extra node is a false positive when compared to authority truth, but it is a true positive when compared to oracle truth.

We now define our version of *precision* against authority and oracle truth as:

$$precision_{authority} = \frac{tp}{tp+overcount} \quad (1)$$

$$precision_{oracle} = \frac{tp+missauth+extra}{tp+missauth+extra+overcount} \quad (2)$$

In general, we do not have false positives (because everything we find is an anycast server). Therefore authority precision reflects our level of accidental overcounts due to multi-server or multiple PR nodes.

Recall captures the coverage of authority truth we obtain:

$$recall = \frac{tp}{tp+fn} \quad (3)$$

We do not define a recall for oracle truth because, by definition, we do not have a complete set of the oracle population.

3.3 Validating Use of CHAOS DNS Queries

We now evaluate our method using CHAOS DNS queries. We know that both of our targets use CHAOS records in their anycast services, so we expect CHAOS records to be very precise, while exercising our recall.

Success of the query infrastructure: We first consider how often our DNS query infrastructure works successfully. Table 1 shows the results of our query process. We see that we are unable to use about 5% (10 or 14 of 242) of our monitoring infrastructure at any given time; typical churn in PlanetLab can be as high as 30% [35]. To reduce the effects of these outages we added second PlanetLab machines at four sites. Our DNS queries require the *dig* program (part of BIND); we were unable to install it on 2 PlanetLab nodes.

About 6% of queries that are sent fail to provide CHAOS records. We see two types of failure. About 5% of queries we send do not reach their destination, suggesting routing churn or server failure. Since we know the servers are professionally run, high-value servers, we assume unreachability represents routing failures. Yet a 5% routing failure rate is 5–25× greater than typical rates of 0.2–1% [22]. This large difference suggests future work to explore whether anycast routing is more prone to failure than unicast routing.

The other 1% of failures for F-root are due missing CHAOS records at a successfully contacted server. Since we expect all servers in this study to support CHAOS records for debugging, our first thought was that these were caused by operator misconfiguration. However, our traceroute method shows that these cases were actually third party nodes masquerading as anycast servers, as discussed in detail in Section 4.1,

Finally, about 94% of the time we get CHAOS records, suggesting the method can provide data almost all the time.

Success of CHAOS result analysis: We next con-

CHAOS DNS Queries:	**F-Root**			**PCH**		
vantage points (duplicates)	242 (4)			242 (4)		
unable to access vantage point	10			14		
DNS tool unavailable	2			2		
query sent	230	100%		226	100%	
no records returned	14	6%	*[100%]*	11	5%	*[100%]*
name server unreachable	12	5%	*[75%]*	11	5%	*[100%]*
no records in response	2	1%	*[25%]*	0	0%	*[0%]*
record returned ($\|\tilde{A}\|$)	**216**	**94%**		**215**	**95%**	

Table 1: Success of our DNS query infrastructure.

CHAOS DNS queries:	**F-Root**		**PCH**
authority truth	47		53
oracle truth	58		53
records considered ($\|\tilde{A}\|$)	216		215
estimated anycast nodes ($\|\hat{A}\|$)	34		26
true positives	21		26
overcounts	12	(0)	0
missing authority	1		0
extra	0		0
authority precision	64%	(100%)	100%
oracle precision	65%	(100%)	100%
recall	45%		49%

Table 2: Accuracy of CHAOS DNS query analysis.

Traceroute:	**F-Root**		**PCH**	
vantage points (duplicates)	242 (4)		242 (4)	
unable to start traceroute	8		14	
traceroute attempted	234	100%	228	100%
no PR	54	23%	27	12%
no ICMP reply	22	9%	21	9%
different target	19	8%	3	1.5%
multiple PRs	13	6%	3	1.5%
PR discovered ($\|\tilde{A}\|$)	**180**	**77%**	**201**	**88%**

Table 3: Success of our traceroute infrastructure.

sider how effective CHAOS DNS queries are at identifying anycast nodes. Table 2 evaluates the process of distilling our observations into estimates. As can be seen, most of our vantage points duplicate coverage.

For coverage, we observe about half of the target's authority truth (recall is 45% and 49% for F-root and PCH). While we have many vantage points, PlanetLab is focused on academic institutions and are biased towards the United States and Europe, while production anycast servers focus on commercial IXPs with broader global coverage. We expect a more diverse set of probing sites would increase our recall. We study the effect of number of vantage points in Section 3.6, and techniques to grow the number of perspectives are future work.

Our precision is better: PCH precision is 100%. F-root precision falls to 64%, mostly because of 12 overcounts. These overcounts are due to T3 or T4 configurations where multiple servers provide service for a single node. Since ISC's CHAOS records are per-server (not per-node), multi-servers result in overcounts.

CHAOS records also reveals one case of incomplete authority truth for F-root. Although missing from the public web page, ISC confirmed that one anycast node we found should have been listed. This missing authority makes our oracle precision very slightly better than authority precision, from 64% to 65%.

Our basic CHAOS algorithm does not interpret the contents of the reply, because there is no formal standard. However, each anycast service provider has their own convention, something we explore in Section 4.3. As an experiment, we decoded ISC's convention, to extract identities of both the anycast node and the specific server. We show the results of this F-Root-aware CHAOS algorithm in parenthesis in Tables 2 and 6. This provider-specific interpretation makes the CHAOS method completely correct, suggesting it would be beneficial to standardize reply contents, or other means of making this distinction.

3.4 Validating Use of Traceroute

We next consider accuracy of the traceroute method described in Section 2.3. An important aspect of traceroute is that its coverage depends on the level of ICMP filtering in the network, rather than on debugging support at the anycast provider.

Success of the traceroute infrastructure: We begin by evaluating how often we can collect data with traceroute. Table 3 shows how often our traceroutes succeed. We define success as traceroute producing a single IP address in both its final and penultimate records. Again, a few PlanetLab nodes are unavailable when we collect our data (8 or 14 of 242, less than 5% of our vantage points).

Many more traceroutes fail than do CHAOS DNS

Traceroute:	**F-Root**	**PCH**
authority truth	47	53
oracle truth	58	53
records considered ($\|\tilde{A}\|$)	180	201
estimated anycast nodes ($\|\hat{A}\|$)	34	33
true positives	18	26
overcounts	13	7
missing authority	2	0
extra	1	0
authority precision	58%	79%
oracle precision	62%	79%
recall	38%	49%

Table 4: Accuracy of traceroute analysis.

queries—23% and 12% for F-root and PCH, compared to only 6% for CHAOS. We find two main causes of these failures. First, in a plurality of cases (22 of 54 for F-Root and 21 of 27 for PCH), routers near the target produced *no ICMP reply* for either the PR or the target. Second, we see two different cases that suggest either multiple interfaces or multiple servers and load balancing (cases T2 and T4 from Figure 2). For *different targets*, the last hop of the traceroute is different than the IP address we are using as our target, likely due to a single server with multiple network interfaces, or multiple adjacent routers on its LAN. For *multiple PRs*, traceroute's multiple ICMP echo requests for the PR return different addresses, suggesting load balancing across multiple machines. Currently we treat these cases as errors, but we believe we can partially interpret them and are in the process of improving our analysis to do so.

Overall, traceroute provides data for most targets (77% and 88%), but slightly less coverage than CHAOS DNS queries. We attribute this difference to the prevalence of ICMP filtering by network operators.

Success of PR analysis: Given traceroute coverage, Table 4 looks at the accuracy of its predictions for our two anycast target services. Compared to CHAOS DNS queries, traceroute finds the same (26 for PCH) or slightly fewer (20 instead of 22 for F-root) true anycast nodes. It also finds many more overcounts: 13 and 7, as opposed to 12 and 0 with CHAOS. F-root operates several multi-server nodes (case T4 in Figure 2), all of which are overcounted by both methods. PCH does not operate multi-server nodes, but may of their nodes have multiple PRs (case T2 in Figure 2) likely due to nodes with multiple interfaces or rich IXP topologies.

These overcounts result in a lower precision for traceroute than for CHAOS queries: 58% and 79% compared to 64% and 100%. We will see that the combination method resolves some of these overcounts.

observations	**F-root**		**PCH**	
vantage points (duplicates)	242 (4)	100%	242 (4)	100%
no CHAOS, unk. PR	17	7%	19	8%
CHAOS and PR	171	71%	193	80%
valid	169	70%	192	79%
false combination	2	1%	1	0.5%
CHAOS or PR, not both	54	22%	30	12%
no CHAOS but PR	9	4%	8	3%
CHAOS but unk. PR	45	18%	22	9%

Table 5: Success of combined infrastructure.

Incomplete authority truth is also a problem. F-root's authority truth is missing two nodes. One of these was found and discussed in CHAOS, the other is found only by traceroute. That anycast node was visible only from a single vantage point, and although it supported traceroute, we were unable to install dig on it to run CHAOS DNS queries. F-root also has one extra node, a third party server operating on the F-root anycast address that we discuss in Section 4.1. These differences in ground truth causes oracle precision to be about 4% higher than authority precision due to different ground truth; we discuss the extra node case in detail in Section 4.1.

Finally, the reduced coverage of traceroute results in a slightly lower recall for F-Root: 38% vs. 45% with CHAOS queries.

3.5 Validating the Combined Method

Our combined method uses both CHAOS DNS queries and traceroute as information sources. It improves results to the extent that those two sources are uncorrelated.

Ability to combine data sources: Table 5 measures the degree of correlation between the two information sources. As can be seen, there is some independence between the two sources, with 22% and 12% of cases present in only one of the two information sources for F-Root and PCH. This independence suggests that a method that combines traceroute and CHAOS records may be more accurate that either of those methods alone. Moreover, in the majority of cases (71% and 80%) for which both data sources are present, a combined algorithm can use the more informative of the two sources, for example, to resolve ambiguities in identifying cases T2, T3 and T5 from Figure 2.

Our combined method introduces one new source of infrastructure failure: if routing changes between the CHAOS observation and traceroute, analysis could incorrectly combine observations from different nodes. We detected these cases and identify them as *false combination* in Table 5, removing them before analysis; they occurred primarily because our prototype took CHAOS and traceroute data hours apart. We plan to take CHAOS observations before and after traceroutes to automate

Combined Method:	**F-Root**		**PCH**
authority truth	47		53
oracle truth	58		53
records considered ($\|\tilde{A}\|$)	225		223
estimated anycast nodes ($\|\hat{A}\|$)	27		26
true positives	21		26
overcounts	3	(0)	0
missing authority	2		0
extra	1		0
authority precision	88%	(100%)	100%
oracle precision	89%	(100%)	100%
recall	45%		49%

Table 6: Accuracy of combined analysis.

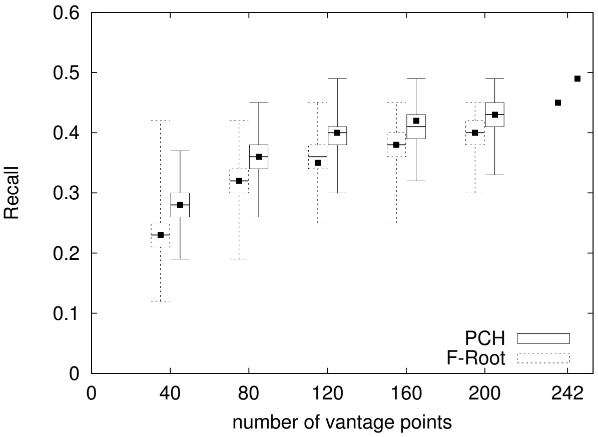


Figure 5: Recall of the combined method as number of vantage points vary. Each box shows median and quartiles, with the whiskers showing extrema. Squares show mean values, or exact results at 242 vantage points. Data for F-Root is on the left of each pair; for PCH is on the right. Exact values (filled squares) represent our best experimental results.

detection of routing changes.

Success of the combined method: Table 6 measures how much our results improve by considering two sources of information. We see that combining sources allows true positives to follow the larger of the two stand-alone methods for both targets. It reduces the number of overcounts by 75% (3 instead of 12 or 13) for F-root, even without decoding F-root CHAOS replies. It eliminates overcounts for PCH.

These improvements translate into better precision and recall for the combined method. For F-Root, precision rises to 88% (compared to 64% or 58% authority precision, with similar results for oracle precision), and PCH precision remains at 100%, the maximum of the single-source algorithms. Recall also follows the maximum of the two algorithms, reaching 45% and 49% of ground truth.

The improvements of the combined method depend on the target. F-Root shows larger improvements because our single-source estimates of it were poorer than were our estimates of PCH. We conclude that the combined method is desirable because it should always improve accuracy.

3.6 Effects of Vantage Point Diversity

Ultimately our recall is dominated by our ability to see different anycast nodes. At best, each vantage point is in a different catchment and sees a new node; at worst, they are all in the same catchment and we are uncertain if the target is using anycast at all. In Figure 4, we see that vantage points a, b and c duplicate each other, as do d and e. This challenge is apparent in the statistics about each method: while the precision of the three approaches varies, the recall of all is similar. Since we find only 40–50% of the anycast nodes of our two trial targets, clearly our vantage points show significant redundancy in their coverage.

Our study of recall is limited by the use of PlanetLab for our measurement platform. For example, F-root has an anycast node in Bangladesh, and PlanetLab's coverage in south Asia is poor, and its nearest node is in India. Unfortunately, it is quite expensive to deploy new vantage points, although we are considering additional sources as possible future work.

To study the effect of the number of vantage points on recall, we can consider subsets of our current measurement infrastructure. Figure 5 shows our evaluation of different size subsets of vantage points from our experimental data. We draw 1000 randomly chosen samples from our observations and plot median and quartiles as boxes, minimum and maximum as whiskers, mean with squares and our maximum recall with squares.

Care must be taken to interpret the results of this experiment; because of the limited size of our vantage point population, results naturally converge as we select large and larger subsets. However, we see that median recall stays fairly high with half the vantage points, but drops fairly quickly with fewer. The best possible recall from the randomly selected subset is quite high even as their number drops; this suggests, as future work, ways to select or deploy the "best" (most diverse) vantage points. Results for both F-Root and PCH are qualitatively similar.

4. EVALUATION

Now that we have validated the accuracy of our approach, we next report three results about anycast use in the Internet. We describe one case where ACE discovered an "extra" anycast node run by a third party, then we evaluate anycast use in both country and generic top-level domains (ccTLDs and gTLDs), and finally we attempt to characterize anycast providers.

4.1 Masquerading Anycast Nodes

As part of our validation of ACE on F-Root, we encountered an anycast server providing root-domain content. However, the PR for this server was not on ISC's list of F-Root anycast nodes, and the CHAOS record returned an empty string. This server was a non-ISC server running on the F-Root IP address.

Discussions with ISC revealed two general cases where other organizations operate nodes at the F-Root anycast address. First, some organizations operate local copies of the root zone, and internally *masquerade* responses to `*.root-servers.org`. While ISC discourages this behavior, redirection of internal traffic is generally left up to the organization. Second, other organizations have attempted to hijack root DNS server from others, often to promote a modified root zone.

For the case that we found, the PR of the target is 202.112.36.246, at AS4538 in CERNET, the China Education and Research Network. We found this anycast responder from two vantage points also inside CERNET: `planetlab-1.sjtu.edu.cn` (202.112.28.98, in the same AS as the anycast node), and `pl1.6test.edu.cn` (219.243.208.60, in AS23910, another AS that whois identifies as run by CERNET). The contents of the two zones appeared the same based on the SOA record, although we did not exhaustively compare the zones. ISC identified this non-ISC anycast node as a masquerading node, not a hijacked one, and we concur.

While this case represents a network operator choosing to handle requests from their own users using masquerading, nearly the same mechanisms can be used for maliciously hijacking DNS servers. This potential illustrates the benefits of actively monitoring anycast services, at least until use of DNSsec becomes pervasive.

4.2 Anycast Use in Top-Level Domains

While we know that anycast is widely used, particularly for DNS servers, to our knowledge there has been no systematic study of *how wide* its use is. We next use ACE to begin to answer this question. Our answer to this question is necessarily incomplete; we know our recall of F-Root and PCH is moderate at best. However, complete identification of *all* anycast nodes for a given domain name is not necessary to begin to understand where anycast is used: knowledge of two anycast nodes for a specific DNS IP address confirm that it employs *some* level of anycast.

Target: Our targets for study are the 227 country-code top-level domain names (ccTLDs), and the 33 generic TLDs (gTLDs). Together they have 1164 name servers at unique IP addresses, 1026 for ccTLDs and 138 for gTLDs. These lists are from the IANA website as of mid-April 2011 [18].

CHAOS	**traceroute** (number of PRs)		
(# recs.)	> 1	1	0
> 1	**anycast**	**anycast**; T3 unicast	**anycast**; T3 unicast
1	T2 unicast; **mis-config anycast**	unicast	unicast; mis-config anycast
0	**non-BIND anycast**; T2/T4 unicast	unicast	insufficient information

Table 7: Interpretation of CHAOS DNS queries and traceroute on TLD nameservers.

gTLD Results				
CHAOS	**traceroute** (# PRs)			**possible**
(# recs.)	> 1	1	0	**anycast**
> 1	**46 (46)**	**1 (1)**	**24 (24)**	**71**
1	**13 (0)**	5 (0)	15 (0)	0
0	**4 (4)**	1 (0)	29 (0)	**4**
			total anycast and unicast:	138 **(75)**

ccTLD Results				
CHAOS	**traceroute** (# PRs)			**possible**
(# recs.)	> 1	1	0	**anycast**
> 1	**130 (120)**	**18 (2)**	**127 (127)**	**249**
1	**177 (1)**	101 (0)	276 (0)	**1**
0	**50 (50)**	44 (0)	103 (0)	**50**
			total anycast and unicast:	1026 **(300)**

Table 8: Possible anycast services discovered for gTLD (top) and ccTLD (bottom) DNS servers. The number in braces is the number, obtained by ACE, of name servers that are likely to use anycast.

Methodology: We apply ACE to each IP address for each name server, recording the CHAOS DNS records and PRs from traceroute. We collected data on April 13th for ccTLD and April 18th for gTLD name servers.

Table 7 shows how we interpret our observations for these TLDs, since we no longer are testing against ground truth. Of these cases $\mathit{CHAOS} > 1 \wedge \mathit{PR} > 1$ is the strongest evidence for anycast, while the other cases where $\mathit{CHAOS} > 1$ or $\mathit{PR} > 1$ are likely partially observed anycast addresses.

Results: Table 8 show our results for gTLD and ccTLD, respectively. We observe that about half of gTLDs nameservers and about one-quarter of ccTLD nameservers show some evidence of anycast use. About

33% (gTLD) or 12% (ccTLD) of all TLD nameservers show strong evidence of anycast with both multiple CHAOS records and multiple PRs. If we adopt $CHAOS > 1 \wedge PR > 1$ as a lower bound and $CHAOS > 1 \vee PR > 1$ as an upper bound, then least 14% (166 of 1164) and perhaps 32% (375 of 1164) of TLD servers use anycast.

The main implication of these findings is that anycast is an important component of the DNS, and needs to be protected from hijack attempts (Section 4.1). Anycast addresses are not handled by traditional route hijack detection methods because hijack detection assumes any multiple routes for the same prefix are a hijack (Section 5), while with anycast, multiple routes are part of regular operation.

4.3 How Many Anycast Providers Exist?

From our observations of top-level domains we next evaluate how many independent anycast systems appear to be operating. Since a number of organizations provide anycast service to multiple parties, our observations from Section 4.2 provide only a loose upper bound.

To identify anycast providers, we reviewed the CHAOS queries to confirmed anycast nodes; we find 240 likely anycast services with $CHAOS > 1$.

To go from services to providers, we examine the patters in their replies. We identify a potential provider based on either a unique pattern, or a provider-specific identifier in a standard pattern. For example, several organizations include operator's domain name in their reply, while others use distinctive patterns. We see two general patterns: in the most common (21 likely providers found), the reply uses hostname-like strings, often encoding geographic information along with server and node identity. Examples of this format include *lax1b.f.root-servers.org* for a server *b* at an IXP-1 in Los Angeles, and *host.bur.example.net* for server *host* at an IXP near Burbank airport. The second format we identified, with 10 likely providers, is even more provider specific, with just a serial number *example1* for server 1 by provider *example*, or server plus a geographic code *s1.lax* for server 1 in Los Angeles.

We found 15 providers serving gTLDs, and 25 providers of ccTLDs. There is some overlap of providers, 31 providers are unique in the entire set. These counts represent likely lower bounds: likely, because it seems unlikely for providers to use very dissimilar patterns, and lower bounds because the second format is general enough that two providers may have adopted the same scheme accidentally.

Finally, Figure 6 shows how many services each provider operates. We can see that most providers are unique to one service (about half, from 18 to 31, are unique). A few large providers operate many services, with the first seven providers operating more than 80% of services (198 of 240).

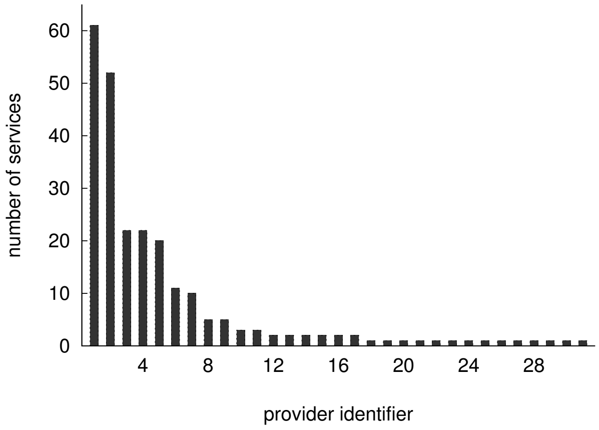


Figure 6: Estimates of number of services (anycast IP addresses) operated by each estimated provider (as identified by CHAOS response patterns).

5. RELATED WORK

While there has been significant work exploring the DNS performance and anycast use in root nameservers, to date there has been little work exploring anycast discovery, at least outside the operational community. We review that work, and broader related work in route hijack detection.

Anycast discovery: The Internet Systems Consortium and the DNS operational community has developed several techniques to support debugging anycast nodes. In RFC-4892 they propose using queries of DNS text records "hostname.bind" and "id.server", where the otherwise obsolete CHAOS class is re-purposed to identify anycast servers [38]. Although originally developed in the BIND implementation of DNS, the approach is now supported in other DNS server software (for example, NSD [26]). We use this approach as one of our methods (Section 2.2).

Subsequent standards activity has suggested the need for additional debugging support. RFC-5001 [3] defines an new NSID (name server identifier) option for DNS. By using a new option, it differs from RFC-4892 in specifying that recursive DNS servers will not forward NSID requests. Although the RFC explores several possible payloads NSID could return, it explicitly defers standardizing contents to future work. A recent draft [23] proposes using unique-per-node AS numbers for anycast node identification. Future anycast surveys can use these methods when they are widely deployed. These approaches suggest the need for additional anycast monitoring tools. We agree more information is needed; our work explores new ways to evaluate anycast with existing services. Our work could be extended as these new methods are deployed.

Root nameserver performance: Several measurement studies have explored proximity and the stability of routing to root and other TLD name servers. Several active probing-based approaches have used PlanetLab nodes [9] and other vantage points to study the affinity and proximity to root servers or several TLD name servers [31, 7, 4, 5, 10]. Most of these approaches used hundreds of vantage points, while Ballani et al. [5] used 20,000 recursive DNS servers. Other studies have attempted to characterize proximity and affinity by collecting data at name servers [6, 21]. Like most of this work, we use PlanetLab to get diverse viewpoints on anycast, but unlike this work we explore anycast structure, not DNS performance.

Two groups have evaluated routing changes to anycast nodes using CHAOS records to discover anycast name servers [8, 32]. Like their work, we use CHAOS DNS queries, but we study anycast structure, not routing.

To our knowledge, to date, no work other than ours has developed a methodology for identifying anycast servers, and then characterized anycast deployment.

Route Hijack Detection: Closest to our study of DNS masquerading and hijacking is work on general route hijacking and protection of routing to name servers.

Before widespread deployment of anycast, Wang et al. [37] proposed a BGP path-filtering method to protect routes to critical TLD name servers. Others have discussed the importance of detecting hijacked *unicast* prefixes [27], and proposed methods for hijack detection using the control plane [20], data plane [40, 39, 29], and hybrid control-and-data plane approaches [16]. Detecting anycast hijacking is qualitatively harder than detecting unicast hijacking, since by definition, anycast packets can be sent to one of many destinations, one or more of which may be suspect while with unicast routing any examples of multiple destinations are illegitimate.

6. CONCLUSIONS

Through its use in many top-level DNS servers, anycast has become an indispensable part of the Internet. We developed ACE, which actively discovers, using several vantage points, anycast nodes by issuing CHAOS DNS queries and traceroutes. We find these approaches have generally good precision: what they find are correct anycast nodes, although depending on the amount of information we can gather, overcounting can affect precision. We study how recall, the completeness of our results, is affected by the number of vantage points. Finally, our studies of F-Root and PCH anycast infrastructure detect one third-party site masquerading as an anycast node, and our evaluation of all country-code and generic top-level domain servers, shows anycast is used by 12% of ccTLDs and 33% of gTLDs.

Acknowledgments

This work would not have been possible without ISC and PCH providing public information about their infrastructure. We thank both organizations for their openness and for their patience answering our questions: Paul Vixie and Leo Bicknell at ISC, and Bill Woodcock and Ross Stapleton-Gray at PCH.

7. REFERENCES

[1] J. Abley and K. Lindqvist. Operation of anycast services. *RFC 4786*, 2006.

[2] Joe Abley. Hierarchical anycast for global service distribution. Technical Report ISC-TN-2003-1, Internet Systems Consortium, March 2003.

[3] R. Austein. Dns name server identifier (nsid) option. *RFC 5001*, 2007.

[4] H. Ballani and P. Francis. Towards a global ip anycast service. In *Proceedings of the ACM SIGCOMM Conference*. ACM, August 2005.

[5] H. Ballani, P. Francis, and S. Ratnasamy. A measurement-based deployment proposal for ip anycast. In *Proceedings of the ACM Internet Measurement Conference*, pages 231–244. ACM, 2006.

[6] P. Barber, M. Larson, M. Kosters, and P. Toscano. Life and times of j-root. In *NANOG 34 meeting*, October 2004.

[7] Peter Boothe and Randy Bush. Anycast measurements used to highlight routing instabilities. In *NANOG 34 meeting*, May 2005.

[8] Randy Bush. Dns anycast stability: some initial results. CAIDA/WIDE workshop, `http://www.caida.org/workshops/wide/0503/slides/050311.wide-anycast.pdf`, March 2005.

[9] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. Planetlab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33(3):3–12, 2003.

[10] L. Colitti. Effect of anycast on k-root. dns-oarc workshop, july 2005. DNS-OARC Workshop, July 2005.

[11] Eric Cronin, Sugih Jamin, Cheng Jin, Anthony R. Kurc, Danny Raz, and Yuval Shavitt. Constrained mirror placement on the Internet. *IEEE Journal of Selected Areas in Communication*, 20(7):1369–1383, September 2002.

[12] John Dilley, Bruce Maggs, Jay Parikh, Harald Prokop, Ramesh Sitaraman, and Bill Weihl. Globally distributed content delivery. *IEEE Internet Computing*, 6(5):50–58, September 2002.

[13] R. Engel, V. Peris, D. Saha, E. Basturk, and R. Haas. Using ip anycast for load distribution and server location. In *Proceedings of the Global Internet*, December 1998.

[14] B. Greene and D. McPherson. Isp security: Deploying and using sinkholes. NANOG talk, http://www.nanog.org/mtg-0306/sink.html, June 2003.

[15] T. Hardie. Distributing authoritative name servers via shared unicast addresses. *RFC 3258*, 2002.

[16] X. Hu and Z.M. Mao. Accurate real-time identification of ip prefix hijacking. In *Proceedings of IEEE Security and Privacy*. IEEE Computer Society, 2007.

[17] C. Huitema. An anycast prefix for 6to4 relay routers. *RFC 3068*, 2001.

[18] Internet Assigned Numbers Authority. Root zone database. http://www.iana.org/domains/root/db/.

[19] Dina Katabi and John Wroclawski. A framework for scalable global IP-anycast (GIA). In *Proceedings of the ACM SIGCOMM Conference*, pages 3–15, Stockholm, Sweeden, August 2000. ACM.

[20] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang. Phas: A prefix hijack alert system. In *Proceedings of the USENIX Security Symposium*, 2006.

[21] Z. Liu, B. Huffaker, M. Fomenkov, N. Brownlee, and K. Claffy. Two days in the life of the dns anycast root servers. In *Passive and Active Network Measurement*, pages 125–134, 2007.

[22] Ratul Mahajan, David Wetherall, and Tom Anderson. Understanding BGP misconfiguration. In *Proceedings of the ACM SIGCOMM Conference*, pages 3–16, Pittsburgh, Pennsylvania, USA, August 2002. ACM.

[23] Danny McPherson, Ryan Donnelly, and Frank Scalzo. Unique per-node origin ASNs for globally anycasted services. *Active Internet-Draft*, November 2010.

[24] P. Mockapetris. Domain names—concepts and facilities. RFC 1034, Internet Request For Comments, November 1987.

[25] David A. Moon. Chaosnet. A.I. Memo AIM-628, Massachusetts Institute of Technology, AI Labratory, June 1981.

[26] NLnet Lab. Name server daemon (NSD). http://www.nlnetlabs.nl/projects/nsd/.

[27] O. Nordström and C. Dovrolis. Beware of bgp attacks. *ACM SIGCOMM Computer Communication Review*, 34(2):1–8, 2004.

[28] C. Partridge, T. Mendez, and W. Milliken. Host anycasting service. *RFC 1546*, 1993.

[29] T. Qiu, L. Ji, D. Pei, J. Wang, J. Xu, and H. Ballani. Locating prefix hijackers using lock. In *Proceedings of the 18th conference on USENIX security symposium*, pages 135–150. USENIX Association, 2009.

[30] root servers.org. Root DNS server website. http://www.root-servers.org/.

[31] S. Sarat, V. Pappas, and A. Terzis. On the use of anycast in dns. Technical Report Tech. rep., HiNRG, Johns Hopkins University Technical Report, 2004.

[32] Yuji Sekiya. Passive and active dns measurement: update. CAIDA/WIDE workshop, http://www.caida.org/workshops/wide/0503/slides/sekiya.pdf, March 2005.

[33] Karen Sollins and Larry Masinter. Functional requirements for uniform resource names. RFC 1737, Internet Request For Comments, December 1994.

[34] Steve Souders. High-performance web sites. 51(12):36–41, December 2008.

[35] Neil Spring, Larry Peterson, Andy Bavier, and Vivek Pai. Using PlanetLab for network research: myths, realities, and best practices. *ACM Operating Systems Review*, 40(1):17–24, January 2006.

[36] The Domain Name System Operations Analysis and Research Center. The as112 project. http://www.as112.net.

[37] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, and L. Zhang. Protecting bgp routes to top-level dns servers. *IEEE Transactions on Parallel and Distributed Systems*, pages 851–860, September 2003.

[38] S. Woolf and D. Conrad. Requirements for a mechanism identifying a name server instance. *RFC 4892*, 2007.

[39] Z. Zhang, Y. Zhang, Y.C. Hu, Z.M. Mao, and R. Bush. Ispy: detecting ip prefix hijacking on my own. In *Proceedings of the ACM SIGCOMM Conference*, pages 327–338. ACM, August 2008.

[40] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis. A light-weight distributed scheme for detecting ip prefix hijacks in real-time. In *Proceedings of the ACM SIGCOMM Conference*. ACM, August 2007.