Measuring the Placement of DNS Servers in Top-Level-Domain

Yingdi Yu
UCLA
yingdi@cs.ucla.edu

Jiwen Cai
UCLA
jwcai@cs.ucla.edu

Eric Osterweil
VeriSign
eosterweil@verisign.com

Lixia Zhang
UCLA
lixia@cs.ucla.edu

ABSTRACT

DNS is a critical infrastructure of the global Internet. To assure DNS's efficient and robust operations, each domain, especially each of the Top-Level-Domains (TLDs), should deploy multiple redundant nameservers in diverse locations. To assess the robustness of TLD nameserver deployment regarding the nameserver redundancy and location diversity, we conduct a measurement study by sending special DNS queries and running traceroute to TLD nameservers from about 200 PlanetLab sites. By combining the measurement data with other data sources including IP address assignment, BGP routing tables, and IP address location inference, we identify the nameserver providers for all TLDs and the locations of their servers. Our results suggest that TLD nameservers are adequately distributed as of today, more than 90% TLDs can survive from single point failure. We also observe that many TLD nameservers are co-located at a small group of "hot spots". We conduct simulations to evaluate the impact of such co-locations. Our results show that failure of single hot spot does not affect robustness of TLD services.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Measurement

Keywords

DNS, Nameserver Placement, Robustness

1. INTRODUCTION

Domain Name System (DNS) is a critical infrastructure of the global Internet. To assure DNS's effcient and robust operations, DNS deployment guidelines [21, 14] suggest that each domain, especially Top-Level-Domains (TLDs), should deploy redundant nameservers in diverse locations.

In this paper we conduct a systematic measurement to assess the robustness of TLD regarding nameserver redundancy and location diversity. However looking from per domain view does not give the complete picture of DNS service robustness. Multiple domains may have their nameservers in the same physical hosts, or place nameservers at the same organizations; the servers provided by different organizations may also share the same geographic locations. Thus we also measure the existence of TLD nameserver co-locations, and evaluate its impact on the overrall robustness of all TLDs services.

A common practice in providing DNS services today is the use of anycast, which represents a major challenge in our measurement efforts. We develop a set of methods to identify providers and locations of DNS nameservers, including anycast nameservers.

The contributions of this work are listed as follows:

- We measure nameserver redundancy and location diversity of each TLD, and assess its robustness against different types of single point failure.
- We measure the existence of TLD nameservers co-location at different granularities, and evaluate its impact on the robustness of all TLDs.
- We present a set of methods to identify providers and locations of nameservers. Our methods can apply to anycast nameservers.

The rest of the paper is organized as follows: Section 2 reviews related works. Section 3 provides background information. We discuss robustness issues related to nameserver placement in Section 4. Section 5 explains the methods to measure providers and locations of nameservers. Section 6 analyzes measurement results. We conclude in Section 7.

2. RELATED WORK

We classify prior related work into three catergories: performance measurement on anycast nameserver, performance measurement on nameserver placement, robustness evaluation on nameserver placement.

Colitti *et al.* [13] evaluated the effects of using anycast on k.root nameserver. Sarat *et al.* [23] used PlanetLab [3] to evaluate performance of f.root, k.root and UltraDNS TLD nameservers. To avoid bias introduced by limited vantage points, Ballani *et al.* [11] utilized open resolvers as their vantage points to evaluate performance of anycast nameservers. As focused on performance measurement, they assumed pre-acquired deployment information of these anycast nameservers. Our work tries to measure the placement of nameservers, including those anycast ones, i.e. we tries to tell which nameservers are anycasted or not, and where their anycast nodes are located.

Lee *et al.* [17] studied the impact of root nameserver placement on DNS performance. They used CAIDA's probing tools at server side to collect data about client-server communication. Brownlee *et al.* [12], and Liston *et al.* [18] used measurement tools at client side to measure performance impact of root and gTLD nameserver placement. In this work, we assess the impact of nameserver placement on service robustness.

Pappas *et al.* [22] listed the lack of diversity in server placements as one of the four types of configuration errors on DNS robustness, and reported that 82% of measured zones placed all their nameservers in the same geographical locations, and 77% of measured zones placed their nameservers in a single Autonomous System (AS). Zones measured by them were randomly selected from a huge collection, and anycast nameservers were not considered. Gibbard [15] reported locations of TLDs and root nameservers in 2006, and discussed potential problems of nameserver deployment. In this work, we report locations of current TLD nameservers five years after Gibbard's work. Moreover, we measure the existence of nameservers co-locations, and evaluate the impact of co-locations on service robustness. We also present a set of systematic methods to measure locations of nameservers, especially for anycast nameservers.

3. BACKGROUND

3.1 Placement Requirements on TLDs

There are two classes of TLDs: country-code TLD (ccTLD) and generic TLD (gTLD). ccTLDs are used to manage names related to corresponding countries or territories, with a few exceptions[1]. Although DNS names under a ccTLD are generally used by entities in that country, it is highly desired for a ccTLD to place some of its nameservers outside its territory to reduce service latency of resolvers in other countries,and to increase service robustness against country-level failures.

DNS names under gTLDs, e.g. ".org", ".net", are usually used by organizations in the global internet community. There are also a few exceptions. For example, ".gov" is limited to U.S. governmental entities and agencies. DNS services for gTLDs are contracted to DNS service providers, such as VeriSign, NeuStart, etc.. Some countries also delegate their DNS services to DNS service providers, or place some of their nameservers at Internet hosting providers.

3.2 DNS Anycast

3.2.1 Existing Anycast Deployment

DNS anycast is a practice of providing DNS service at multiple geographic and topological locations [16, 7]. Physical servers at different locations use the same IP address, but a DNS query sent to that IP address can be routed to only one location. Which location a DNS query is routed depends on the routing protocol in use, and is usually the closest one in terms of topology.

Anycast DNS servers at different locations announce the same IP address prefix to neighbor ASes. Operators of an anycast nameserver may localize its service by advertising a longer prefix within a local routing scope, e.g. by setting the no-export attribute in BGP announcements, so that local DNS resolvers, and only local DNS resolvers, will use this local anycast server. We call such servers *local nodes*. Majority of anycast DNS servers are globally scoped, i.e. their routing reachability announcements propagate globally. We refer to such servers as *global nodes*.

3.2.2 Existing Methods of Identifying Anycast

There are two existing methods to identify anycast nodes at different locations: nameserver ID, and origin AS number.

One method requires anycast nameservers to provide its identity information. These identity information can be fetched by a special DNS query [24, 10]. For example, we can query hostname.bind record on machines using BIND [8]. This method can provide accurate identity information, however its usefulness requires support from DNS operators. Unfortunately, our measurements show that about 2/3 TLD nameservers do not provide their identity information. Moreover, we cannot directly infer location of an anycast nameserver from its identity information.

Some works suggested unique origin AS number per anycast node as another identifying method [20]. However, a number of existing anycast deployments use a single AS number for routing announcements from all the anycast nodes. As a result, we cannot rely on the origin AS number to identify anycast nodes.

We develop a set of methods to identify individual

[1] For example, ccTLD ".la" is supposed to be used for names in Lao, but it actually manages many names related to Los Angeles.

instances of anycast DNS servers using a combination of hostname and traceroute data. We will discuss them in detail in Section 5.

4. PLACEMENT AND ROBUSTNESS

In this section, we discuss potential issues of nameserver placement that can affect robustness of DNS services. To facilitate discussion, we first define related terminologies and failure models.

4.1 Terminology Definitions

4.1.1 Server Node

An anycast nameserver can have multiple physical servers at different locations. At each location, multiple physical servers can provide the same DNS service as a cluster. We define such a cluster of physical servers as a *server node*. With this definition, an anycast nameserver can be viewed as nameserver with multiple server nodes. Server nodes with locally advertised routing announcement are local server nodes. Server nodes with global routing announcement are global server nodes. The definition can also apply to unicast nameserver, which can be viewed as nameserver with only one global server node.

4.1.2 DNS Service Provider

As mentioned in Section 3.1, some domains delegate their services to some DNS service providers. A domain can place all its nameservers at one DNS service provider to facilitate management. It can also distribute its nameservers to multiple DNS service providers to achieve location diversity at provider level. A DNS service provider can deploy multiple nameservers for a domain. If a nameserver is anycast, all its server nodes belong to the same DNS service provider.

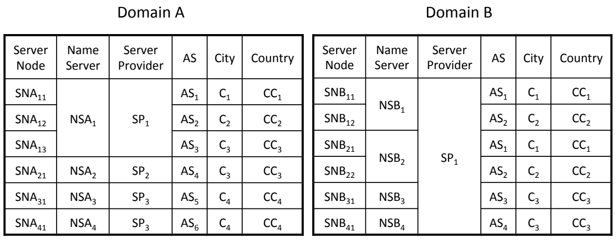
4.1.3 Server Provider

If some domains do not want to delegate their services to DNS service providers, they can place their nameservers at some Internet hosting providers to achieve location diversity. Some domains can have some nameservers at DNS service providers, and some other nameservers at Internet hosting providers. As we focus on how many parties provide servers for a domain, we do not distinguish if a party is DNS service provider or Internet hosting provider. Therefore, we define *server provider* to represent both of them.

4.2 Failure Models

Before we define specific failure models, several failure relatinships should be clarified.

Given a server node with multiple physical servers, it will fail when services on all its phyiscal servers can not be accessed.



Domain A

Server Node	Name Server	Server Provider	AS	City	Country
SNA_{11}	NSA_1	SP_1	AS_1	C_1	CC_1
SNA_{12}			AS_2	C_2	CC_2
SNA_{13}			AS_3	C_3	CC_3
SNA_{21}	NSA_2	SP_2	AS_4	C_3	CC_3
SNA_{31}	NSA_3	SP_3	AS_5	C_4	CC_4
SNA_{41}	NSA_4	SP_3	AS_6	C_4	CC_4

Domain B

Server Node	Name Server	Server Provider	AS	City	Country
SNB_{11}	NSB_1	SP_1	AS_1	C_1	CC_1
SNB_{12}			AS_2	C_2	CC_2
SNB_{21}	NSB_2		AS_1	C_1	CC_1
SNB_{22}			AS_2	C_2	CC_2
SNB_{31}	NSB_3		AS_3	C_3	CC_3
SNB_{41}	NSB_4		AS_4	C_3	CC_3

Figure 1: Example of two domains' nameserver deployment

Given an anycast nameserver, we do not distinguish failure impacts of its global nodes or local nodes. When a node fails, BGP routing can redirect queries to other available global nodes. If there is no available global node, people can use other methods to get reply from rest local nodes, such as querying other resolvers that can reach those local nodes. Thus, an anycast nameserver will faill when all of its server nodes can not be accessed. This aslo apply to unicast nameserver, which will fail when its only node fails.

Given a domain with multiple nameservers, we define its failure as none of server nodes belonging to its nameservers can be accessed. For a unicast nameserver, we count it as one server node. For a anycast nameserver, we count it as multiple server nodes, each server node corresponds to one location where it service is provided. Such a definition allows us to use the number of server nodes in a domain to evaluate its redundancy.

Next, we list several types of failures that can lead to failures of a set of server nodes at the same time:

- **Server Provider Failure**: Operational errors, or network equipment failures may break down a server provider's network. In these cases, all server nodes of the provider will fail to provide DNS services.
- **AS Failure**: Failures of an AS can make all server nodes in it unreachable.
- **City Failure**: A city may experince unexpected accidents or disasters, such as black out, earthquake. In these cases, all server nodes located in the city will fail to provide DNS services.
- **Country Failure**: The internet access of a country may be blocked. In this case, all server nodes inside the country will be isolated from the rest of world.

We illustrate the impact of each types of failures in an example as shown in Figure 1. Figure 1 shows two domains' nameserver deployment. Domain A has 4 nameservers placed at 3 server providers, among which SP_3 provides 2 nameserver: NSA_3, NSA_4. Among 4 nameservers, NSA_1 is anycast. It has 3 server nodes, so there are 6 server nodes for domain A. These servers locations are also shown in Figure 1. Another domain B has also has 4 nameservers, but all of them are placed at 1 server provider SP_1. Among 4 nameservers, NSB_1 and NSB_2 are anycast. Each of them has 2 server nodes, so there are also 6 server nodes for domain B.

Both domains can afford failures of 5 server nodes. They can also afford failures of 3 nameservers. However, impacts of server provider failure are quite different on the two domains, due to their different diversities on server providers. Domain A can afford failures of 3 server providers, but domain B can not afford failure of its only server provider. We can also see that domain A places its server nodes into 6 ASes, 4 cities and 4 countries, but domain B places its server nodes into 4 ASes, 3 cities and 3 countries. Therefore, the number of failures they can afford at each granularities are also different from each other.

4.3 Location Diversity of Nameserver

In order to assess the robustness of TLDs, location diversities at those granularities mentioned above are measured to examine if they can survive from single point failure of each types listed in Section 4.2.

There are two approaches to achieve location diversity of nameservers. One is deploying multiple unicast nameservers at different server providers, different topological locations and geolocations. This approach limits the maximum number of nameservers in a domain to be 13. The other one is deploying anycast nameservers. This approach allows deploying more than 13 server nodes for a domain. However, an anycast nameserver is hosted by one server provider, all of its server nodes may suffer from single point failure of server provider.

Compared with gTLD, ccTLD's service is more accessed by clients in its own country. It is unnecessary and expensive for some countries to anycast their ccTLD services. It is more practical to place a few unicast nameservers outside their territories as a backup in case of country level failure, or to delegate some nameservers to server providers that can provide global accessibility with their own anycast servers.

4.4 Co-location of Nameserver

By hosting more domains' nameservers, commercial DNS service providers can save costs and increase profits. On the other hand, some server providers are preferred by many domains in hosting nameservers due to various reasons, e.g. rich experiences in service provisioning, ability to provide anycast services. In order to facilitate management and reduce cost, some server providers may host multiple domains' nameservers in the same physical servers. Some cities and ASes have rich network connectivities. Consequently, multiple server providers choose the same locations to place their servers.

These factors mentioned above lead to some "hot spots" at different granularities (e.g. physical servers, providers, locations). If many domains place all their nameservers at the same set of "hot points", we can not afford failures of these "hot spots", even though location diversity of each domain looks good. Therefore, impact of these "hot spots" failures should be evaluated.

A special case should be noted here: although nameservers provided by some providers are assigned different IP address, they may be actually hosted in the same server node. We detect this case by querying nameserver IP of one domain (e.g. domain A) for SOA RR of other domains (e.g. domain B). If two domains use separate nameservers, nameserver of domain A will not reply SOA RR of domain B. However, in some case we do get the desired SOA RR from unrelated nameserver IP address. We highly speculate these domains' nameservers are hosted in the same server node. For example, RIPE hosts 75 ccTLDs' nameservers with 75 different IP addresses, but we can get SOA RRs of all the 75 ccTLDs by querying any of the 75 IP addresses. Our speculation has been confirmed by operators in RIPE. We find such cases are prevalent in many large server providers, e.g. Netnode, NeuStar. However, lack of confirmation from them, we still take these nameservers as separate in our measurement.

5. MEASUREMENT METHODOLOGY

To examine the placement issues mentioned in Section 4, we measure the redundancy and locations of nameservers for each TLDs. Given a nameserver, we measure: 1) its server provider, 2) its deployment type, i.e. anycast or unicast, 3) the number and location(s) of its server node(s). Identifying server providers is trivial. We explain how to solve the last two problems in this section. We develop a set of methods that combine data from different sources to infer possible anycast nameservers and their server nodes.

5.1 Data Source

Our measurement collects three types of data: IP addresses of all TLD nameservers, identity information of nameservers, and data of tracerouting from multiple vantage points to nameservers. Our methods also involve three external data sources: IP address assignment, BGP routing and IP location inference.

5.1.1 Data Collection

In order to collect IP addresses of all TLD nameservers, we download root zone files from b.root everyday. To avoid impact of delegation inconsistency and lame delegation[22], for each TLD, we query all its nameservers listed in zone file to find out nameservers not covered by root zone file. Moreover, We also verify the authority of nameservers by querying the TLD's

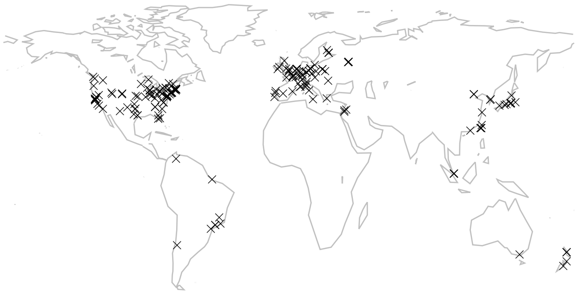


Figure 2: Location of 186 vantage points

Table 1: Number of countries covered by 186 vantage points and number of vantage points in different continents

Continent	Country	Vantage Point
N. America	2	85
Europe	17	59
Asia	8	26
S. America	3	11
Australia	2	5

SOA RR and checking the "aa" bits [21] in reponses. In this way, we find 1143 TLD nameserver IP addresses.

As we mentioned in Section 3.2, query from one location can only be routed to one anycast node. It requires a large set of vantage points in both topologically and geographically different locations to find as many anycast nodes as possible. We use PlanetLab sites [3] as our vantage points to send traceroute probes and DNS hostname queries. Some PlanetLab sites are not stable enough for our measurement, some are hosted in networks which block ICMP traffic. We filter out those unstable and "ICMP-blind" sites, and get 186 useful sites remaining. These sites are located in 32 countries of 5 continents. There is no vantage point in Africa because the there are too few PlanetLab sites in Africa and they are not stable enough to support our measurement. Figure 2 and Table 1 show the geographical distribution of the selected sites.

From all 186 PlanetLab sites, we send out DNS queries for hostname.bind and traceroute probes targeting nameservers of all TLD and the root zone. Two tools are used: scamper [19] in ICMP-paris model can generate special ICMP echo requests to mitigate unstable routing caused by load balancing on router [9]; and rdnsD [4], deployed at each PlanetLab site, is used to reflect DNS queries sent from a centralized controller to destined nameservers. All probes start to be sent out on 8:00 AM UTC from all vantage points everyday. Meanwhile, in order to avoid multiple vantage points sending out ICMP probes to the same target at the same time, the order of targets to be queried is randomized in different vantage points.

5.1.2 External Data Source

In order to identify server providers, we acquire IP address assignment data from EyeP [1], which collects and integerates IP address assignment fetched from five Regional Internet Registries (RIRs). Most server providers usually have their own IP blocks, and can be identified by looking up the owner of IP blocks which nameserver's IP address falls in. It should be noted that the accuracy of our results depends the *whois* database of those RIRs.

To identify ASes which a nameserver is in or peers with, we acquire the data of mapping between IP prefix and its origin ASes from IRL's Internet Topology Collection [6], which collect BGP routing data from thousands of router.

To identify geographical locations where a nameserver is located, we use Maxmind's GeoCityLite databases [2] to acquire data about IP address location inference.

5.2 Identifying Anycast

We develop a set of methodologies to identify anycast and figure out geographical locations and topological locations of nameservers. Our methodologies include 3 steps: *Traceroute path-merging*, *Anycast node inference* and *Location inference*. We explain each step in this section.

5.2.1 Traceroute Path Merging

Traceroute paths may be incomplete, because some intermediate routers may block ICMP traffic, or do not respond to ICMP echo requests. An incomplete traceroute path may introduce false last reachable IP address. However, in our data, most imcomplete traceroute paths end at some router in a complete traceroute path to the same destination, because the ISP balances the traffic at that router, and enforce the ICMP blocking policy at some balanced routers next to the balancing one. Figure 3 shows an example for better understanding, router B balances traffic, router D enforces ICMP blocking. ICMP packets to the same destination are sent from two sites, but are distributed to routers C and D. Packets will be droped by D, but the others can reach the destination through C. The traceroute paths observed at the two vantage points are (..., A, B, *, *, ...) and (..., A, B, D, E, ...), but actually, they can be merged as one, i.e. (..., A, B, D, E, ...).

Such an observation can help us to mitigate impact of incomplete traceroute paths by identifying the common part between complete traceroute paths and incomplete ones.

After merging traceroute pathes, we collect distinct last reachable IP addressses. Anycast nameserver usually have multiple IP addresses for the last reachable

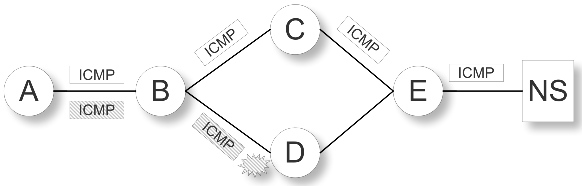


Figure 3: Example of ICMP blocking with load balance

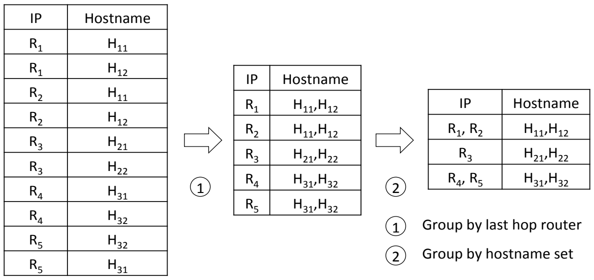
router in front of its anycast nodes. If all vantage points reach a nameserver through the same last reachable IP, the nameserver can be identified as unicast. For rest nameservers, we can group last reachable routers by hostname data (Section 5.2.2) or by other information, such as IP prefix and origin ASes (Section 5.2.3), depending on whether hostname data is available.

5.2.2 *Grouping with Hostname*

It requires multiple hostname.bind values to group last reachable IP addresses. Since some operators may set the same hostname on all their physical servers, nameservers with only one hostname will be treated as those without hostname record. Their last reachable IP addresses should be grouped using methods described in Section 5.2.3.

Multiple hostname values are strong evidences of anycast. However, multiple physcal servers at one anycast node may have different hostname values. For example, f.root has an anycast node at Palo Alto with two physical servers. Each of them has a different hostname: "pao1a.f.root-servers.org" and "pao1b.f.root-servers.org" respectively. They should not be taken as two anycast nodes. It seems easy for human to match those two hostnames to one anycast node. However, it is difficult for computer to group them according to semantics in hostname, because operators of different anycast nameservers may have their own conventions of setting hostname, e.g. "s1.***" for i.root, and "M-***-*" for m.root. To solve this problem, we use last reachable IP as supplementary information to group similar hostnames, as shown in Figure 4,

The idea is based on observation that queries to such an anycast node are load balanced to its physical servers. In most cases, all physical servers can be reached through the same last hop router. Therefore, we can first group hostname by last reachable IP. The results, i.e. multiple sets of hostnames, contain all hostnames of physical servers can be reached through each last reachable IP. Since each hostname set represents an anycast node, we group last reachable IP by the same hostname set. Thus we can obtain a set of last reachble routers' IP for each anycast node, named router set. As we can reach the same anycast node through any elements in a router set, we select one element as a representative for the anycast node in location inference.



IP	Hostname
R_1	H_{11}
R_1	H_{12}
R_2	H_{11}
R_2	H_{12}
R_3	H_{21}
R_3	H_{22}
R_4	H_{31}
R_4	H_{32}
R_5	H_{32}
R_5	H_{31}

IP	Hostname
R_1	H_{11},H_{12}
R_2	H_{11},H_{12}
R_3	H_{21},H_{22}
R_4	H_{31},H_{32}
R_5	H_{31},H_{32}

IP	Hostname
R_1, R_2	H_{11},H_{12}
R_3	H_{21},H_{22}
R_4, R_5	H_{31},H_{32}

Figure 4: Example of ICMP blocking with load balance

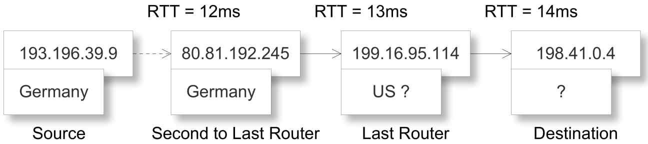


Figure 5: Detecting global distributed IP block

5.2.3 *Grouping without Hostname*

For those nameservers that return one or no hostname, we use a heuristic-based method to group multiple last reachable IP addresses by multiple types of information.

There are two cases that we can observe multiple last reachable IP addresses: 1) Probes may arrive at the same destination, but through different last hop routers; 2) Probes may arrive at the same last hop router, but through different interfaces. In many cases, these IP addresses before an anycast node are in the same IP block, i.e. IP prefix. In some other cases, these IP addresses are from different IP blocks, but their prefixes are announced by the same origin AS. Thus, we can use IP prefix and origin AS to group last reachable IP addresses.

However, since our data source of IP location inference reports location at granularity of IP block, the heuristic may fail when operators assign IP addresses in the same IP block to routers in front of anycast nodes at different locations. We use *globally distributed IP block* to refer this type of IP block. We find such a IP block in measuring a.root. Addresses in IP block 199.16.95.0/24 are assigned to last hop routers for two nodes in United States and one in Europe. But as reported by Maxmind, all these IP addresses are located in Sterling, VA, United States. We try to detect these global distributed IP blocks by checking inconsistency between RTT value reported by traceroute and physical distance according to data of Maxmind. For example, Figure 5 shows partial traceroute path from a vantage point in Germany to one anycast node of a.root. According to Maxmind, the last IP is located in United States and the penultimate IP is in Germany. Nevertheless, the difference between the two routers's RTT is around 1 ms, it is too short for packets to travel between routers across the Atlantic Ocean. Thus, for each traceroute path, we check the RTT of last five hops. Once such inconsistency is detected, we will take the previous router's address (e.g. 80.81.192.245 in Figure 5) as the last reachable IP address instead.

5.2.4 *Location Inference*

After identifying which namservers are anycast or not, we can identify their locations in different ways. For unicast nameservers, we use their own IP addresses to find their geographical locations in GeoLiteCity database provided by Maxmind, and hosting ASes by perform longest prefix match in IRL's Internet Topology Collection [6]. For anycast nameserver, we use the same methods, but with last reachable IP address instead of nameserver IP address.

Previous work [25] pointed out that mapping traceroute path into AS path can be error-prone. The major reason is that border routers use IP addresses belonging to a neighbor AS. To estimate the impact of such a mismatch on our measurement, we checked 212,598 traceroute pathes from 186 vantage points towards 1143 destinations. Among these traceroute pathes, 29% pathes have last reachable routers at border of two different ASes. Location inference based on the rest 71% pathes does not suffer from inaccurate IP2AS probelm. AS to the 29% last reachable routers at the borded of two ASes, their IP addresses may belong to a neighbor AS, which may compromises the accuracy of our topological location inference.

5.3 Methodology Validation

In order to validate our methodology, we use the root nameservers as test cases, because they provide detailed deployment information, and their deployments are diverse, especially on anycast [5]. We also contact operators of root servers to get information up-to-date. The deployment information is taken as ground truth to evaluate the accuracy of our inference. It should be pointed out that all root servers have hostname.bind configured, however, some TLD nameservers do not support that. Thus, we validate our methodology in two tests: one uses all collected data including hostname.bind to see the best result we can get; the other one uses partial data without hostname.bind to see the accuracy of our methods using only traceroute data. Results indicate that, although accuracy of our methods without hostname.bind will be affected a little, our methods can still provide trustable results. In this section, we present the validation results, and explain factors that would affect the accuracy of our methods.

Table 2: Measurement results on root servers with hostname.bind data

Name Server	Operator	Number reported	Number detected
a.root	Verisign	4	4
b.root	ISI	1	1
c.root	Cogent	6	6
d.root	U of Maryland	1	1
e.root	NASA	1	1
f.root	ISC	48	18
g.root	DoD	6	4
h.root	US Army	1	1
i.root	Netnod	36	19 (22)
j.root	Verisign	54	22
k.root	RIPE	18	12 (11)
l.root	ICANN	39	14
m.root	WIDE	6	4

5.3.1 *Validation with Hostname*

We list results of the test using hostname.bind in Table 2. The third column shows numbers of anycast nodes reported by operators. The fourth column shows numbers of anycast nodes measured with our methods. We can see that 4 unicast root servers (i.e. b.root, d.root, e.root, h.root), can be identified by our methods. For rest 9 anycast root servers, we can detect about half of their anycast nodes. Four factors constrain us to detect all anycast nodes correctly: locations of vantage points, number of vantage points, unstable routing, and unauthorized anycast nodes.

Inadequate of location diversity of our vantage points accounts for most missing anycast nodes. Detecting all locally visible nodes raises a very rigorous requirement on locations of vantage points. For example, our measurement missed 7 local nodes of k.root. Among these 7 nodes, 6 are located in countries without PlanetLab sites. The rest missing node is located in Russia. Although we have PlanetLab sites in Russia, they may not reach the anycast node due to routing policies. We believe that increasing location diversity of PlanetLab deployment, or using other more diverse platforms can help us to find out more anycast nodes.

Limited number of vantage points is another factor that can affect accuracy of our methods. It will introduce false positive results. Table 3 shows such an example in k.root, two different hostname.bind values are returned. Traceroute data and hostname indicate that there are two different physical servers with different last hop routers. However, according to k.root's operator, both two hostanames belong to one anycast node in Germany. Since only two PlanetLab sites can reach

Table 3: An example of false positive inference on k.root

Hostname	Last Hop Router
k1.denic.k.ripe.net	80.81.192.154
k2.denic.k.ripe.net	81.91.160.97

Table 4: An example of anycast node inference on i.root

Hostname	Last Hop Router
s1.pix	129.250.11.58
s1.sth	194.146.105.187

them, there is not enough information for our methods to group them as one anycast node. Such cases are very rare. We could expect that our methods can perform better with more PlanetLab sites that can reach the node.

As described in Section 5.2.2, we group hostname.bind by last reachable IP address. It requires vantage point to fetch hostname.bind record from the same anycast node reached by traceroute probes. Although we send out both types of probes at the same time, unstable routing would lead to inconsistency, and introduce false negative results. For example, in Table 4, we list 2 of i.root's anycast nodes and their last hop routers. On April 27th, some vantages points fetched hostname.bind "s1.pix", but their traceroute requests reached the one with "s1.sth". When grouping hostname.bind by last reachable IP, "s1.sth" and "s1.pix" were mistaken as hostname.bind of two machines at the same anycast node. This case also happens very rarely. And it can be filtered out by inspecting data collected in a long duration.

Last factor that can affect our results is unauthorized anycast nodes. Our results indicate that 12 root servers (except h.root) have anycast nodes in China. However, according to operators' reports, only f.root, i.root, and j.root have deployed anycast nodes in China. In our measurement, China is the only country running unauthorized servers. Data in Table 2 does not include unauthorized nodes in China.

5.3.2 Validation without Hostname

We list results without using hostname data in Table 5, and compare them with results using hostname data. There are four types of factors that can cause errors: 1) Traceroute blocking, 2) Globally distributed IP block, 3) Multiple nodes peering the same AS, and 4) Node at Internet eXchange Points (IXP). We will discuss each factor as follwoing.

Table 5: Measurement results on root servers without hostname.bind data

Name	With	Without	False	Causes
Server	Hostname	Hostname	cases	
a.root	4	12	$8^{\ddagger}$	2)
b.root	1	1	0	-
c.root	6	5	$1^{\dagger}$	2)
d.root	1	1	0	-
e.root	1	1	0	-
f.root	18	30	$12^{\ddagger}$	2)
g.root	4	N/A	N/A	1)
h.root	1	1	0	-
i.root	19 (22)	22	0	-
j.root	22	23	$3^{\dagger\ddagger}$	2)
k.root	12 (11)	16	$4^{\ddagger}$	4)
l.root	14	15	$4^{\dagger\ddagger}$	3), 4)
m.root	4	6	$2^{\dagger}$	3)

Our methods will fail if some networks intend to hide from traceroute, e.g. DoD's network where g.root is located. Fortunately, among all TLD's nameservers, only 8 servers are located in network like DoD's. Six of them belong to gTLD ".mil" which is hosted in DoD network as well. The other two belong to ccTLD ".mp".

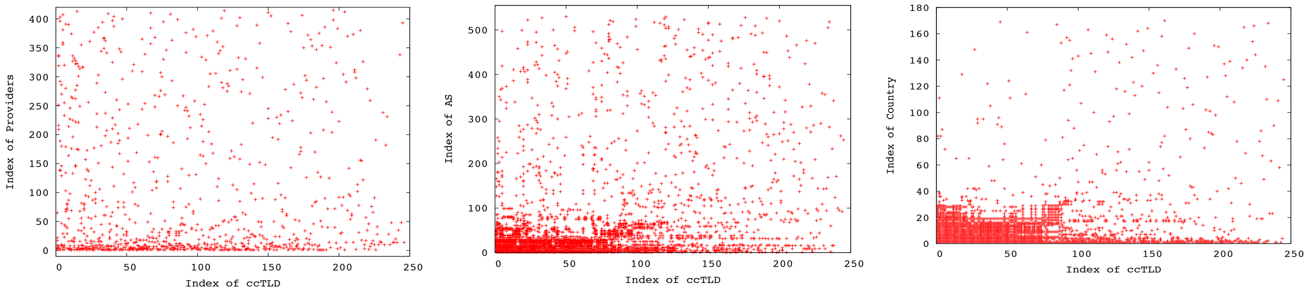
When grouping last reachable routers by IP prefix, our methods may mistake routers in front of different anycast nodes as they are in front of the same anycast node, if these routers use IP addresses in the same globally distributed IP block. Although we can mitigate falsely grouping some of these routers by checking inconsistency of their RTT value as discussed in Section 5.2.3, this technique does not apply to routers without obvious inconsistent RTT values (e.g. two anycast nodes of c.root in Europe), so we may get false negative results in this case. On the other hand, some routers in front of the same anycast node may also have inconsistent RTT (e.g. some anycast nodes of a.root). Our method may mistake them as reaching different anycast nodes, this leads to false positive results. Among 1143 TLD nameserver, only 95 nameservers have this problem, but 60 of them are configured with adequate hostname.bind. Therefore only 35 (3%) nameservers are affected by this problem.

Some operators may deploy multiple anycast nodes peering with the same AS. For example, l.root has three instances in Australia peering with the same AS 7575. When grouping last reachable routers by origin AS of their IP addresses, these anycast nodes may be mistaken as the same one. We may get false negtive results in such a case.

Some anycast nodes deployed at IXP may also introduce false positive. For example, m.root has one anycast node in Paris, connected to at least three ISPs

†false negative results

‡false positive results

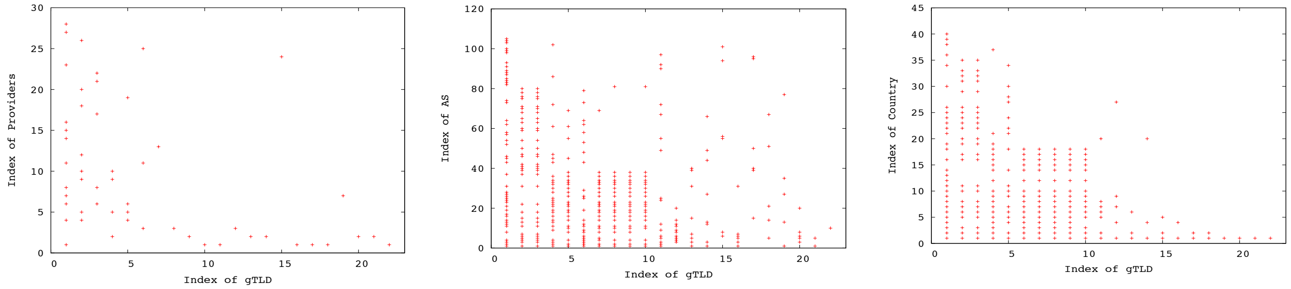


(a) Server Provider

(b) ASes

(c) Countries

Figure 6: Distribution of ccTLDs on countries, ASes, and server providers



(a) Server Provider

(b) ASes

(c) Countries

Figure 7: Distribution of gTLDs on countries, ASes, and server providers

(Renater, France Telecom, and TISCALI). Our method may report two false positive nodes in this case.

The problem of multiple instances peering with the same AS and the problem of single instance at IXP contribute 10 false cases out of 109 anycast nodes in root zone.

5.3.3 *Overall Effectiveness of Methodology*

We summarize the overall effectiveness of our methodology in measuring TLD nameserver placement. Among 1143 TLD nameservers, 341 nameservers have only one unique last reachable IP, and can be confidently labeled as unicast. By grouping with hostname (Section 5.2.2), we detect 171 nameservers which have multiple hostnames but are possibly unicast. By grouping with other information (Section 5.2.3), we detect 161 nameservers with multiple last reachable IP addresses but are possibly unicast. The remaining 470 nameservers are labeled as anycast. Among 470 anycast nameservers, 222 nameservers do not have available hostname.bind record. Grouping last reachable IP without hostname may reduce the accuracy of anycast identification result. However, its impact is litmited to these 222 nameservers.

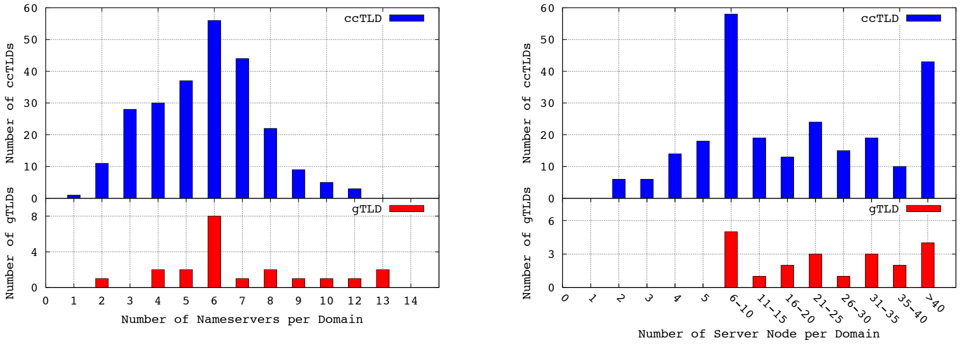
According to validation on root servers, the dominant limitation of our methods is the inadequate number of vantage points, so that our methods may not measure as many server nodes as deployed. With this in mind, readers should be aware that the DNS architecture is at least as robust as we reported in the next section.

6. RESULT AND ANALYSIS

We analyze the placements of nameservers from two aspects: robustness of each TLD regarding its nameserver placement is examined; the impact of nameserver co-location is evaluated by conducting two sets of carefully designed simulations.

Figure 6a and Figure 7a show the distribution of ccTLDs and gTLDs on server providers perspectively. Each point indicates that a domain places at least one nameserver at one server provider. TLDs are indexed according to their numbers of server providers. The one with the most server providers has the smallest index. Similarly, server providers are also indexed according to numbers of server nodes they host. The one with the smallest index hosts the most server nodes of ccTLDs and gTLDs respectively. Note that, although some server providers may host a lot of server nodes, some of these server nodes may support the same TLD. Some server providers with multiple server nodes may not support corresponding numbers of TLDs. Figure 6b and Figure 7b show the distribution of ccTLDs and gTLDs on ASes. Figure 6c and Figure 7c show the distribution of ccTLDs and gTLDs on countries.

Looking Figure 6 and Figure 7 along x-axis can reveal



(a) **(b)**

Figure 8: Distribution of TLDs on: (a) number of NS RRs; (b) number of server nodes

each domain's location diversity at different granularities. While looking along y-axis, we can discover "hot spots" at different granularities and which TLDs node they support. We first examine placement of redundant nameservers in gTLD and ccTLD respectively.

6.1 Placement of Redundant Nameservers

6.1.1 Placement of gTLDs' Nameservers

About 60% gTLD nameservers are anycast. Among all 630 gTLDs' server nodes, 91% are anycast server nodes. Figure 9 shows that only two gTLDs do not have anycast nameserver. The gTLDs are ".xxx" and ".name". ".xxx" is new gTLD that became operational since April, 2011.

Anycast significantly increases nameserver redundancy of gTLDs. As shown in Figure 8b, more than 72 % gTLDs deployed more than 13 server nodes, although DNS specification limits the maximum number of NS RRs of a domain to be 13. All gTLDs have deployed at least 6 server nodes. This indicates that all gTLDs can afford failure of 5 server nodes at the same time.

Next, we examine the location diversity of server nodes at three granularities, as shown in Figure 10. As a result of widely deployed anycast nameserver, nameservers of most TLDs are diversely placed at toplogical and geographical locations. Figure 10b shows the distribution of TLDs on number of their hosting ASes. We can see that 21 gTLDs place their server nodes into multiple ASes, so these gTLDs can survive from single point failure of ASes. The only one exception is ".xxx". Since ".xxx" is new gTLD, it is reasonable that its current nameservers placement does not have so much location diversity.

Figure 10c shows the distribution of TLDs on number of their hosting countries. Only 4 gTLDs place their nameservers inside one country. They are ".edu", ".gov", ".name" and ".xxx", among which ".gov" and ".edu" are actually only used by U.S.

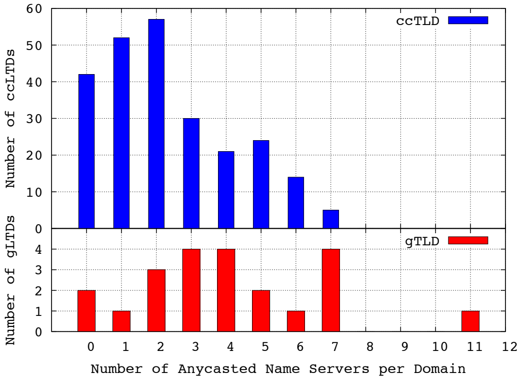


Figure 9: Distribution of TLDs on number of their anycast nameserver.

However, gTLDs' nameserver placement does not exhibit diversity on server providers, as shown Figure 10a. 15 gTLDs (70% of 22 gTLDs) place all their nameservers at one server provider. These gTLDs may be unable to provide DNS services when their server providers fails. The gTLDs with multiple server providers are: ".arpa", ".aero", ".cat", ".int", ".museum", ".tel".

To sum up, for most gTLDs, except ".name" and their new member ".xxx", their nameservers are diversely placed at different locations. However, since most gTLDs palced all their nameservers at only one server provider, they may suffer from such a vulnerability.

6.1.2 Placement of ccTLDs' nameservers

42 ccTLDs (17% of 246 ccTLDs) only use unicast nameservers, but rest 204 ccTLDs have at least one anycast nameserver. This is because many countries delegate their nameservers to server providers that support anycast services. These server providers, such as Afilias, host multiple ccTLDs' nameservers in a set of its anycast nameservers.

Anycast does help increasing nameserver redundancy of ccTLDs. Comparing Figure 8a and Figure 8b, we can find that all bars are shifted to right. ccTLD ".td", the one with only one valid NS RR, have multiple server nodes now. Thus, ccTLDs do not suffer from single point failure of server node. And more than 95 % ccTLDs can afford failures of at least 3 server node at the same time.

Next, we examine the location diversity of ccTLD's nameserver placement. In Figure 10a, we can see that only 19 ccTLDs (8% of all ccTLDs) delegate all their nameservers to one server provider. In Figure 10b, only 8 ccTLDs (3% of all ccTLDs) place all their nameservers in one AS. And only 12 ccTLDs (5% of all ccTLDs) put all their nameservers inside one country. These imply that 92 % ccTLDs can survive from single point failure of server provider, 97% for single point failure of AS, and 95% for single point failure of country.

The nameserver placement at granularity of countries desires further discussion. In our results, among 12 ccTLDs placing all nameservers in one country, 6 ccTLDs (2% of all ccTLDs) put all nameservers inside their own countries. When these countries block their network, people outside can not access services of their ccTLDs. On the other hand, 80 ccTLDs (33% of all ccTLDs) do not host any of their nameservers inside their own countries. These countries can not afford losing network connectivity to the world outside. In that case, people in their countries can not access services of their own ccTLDs.

6.2 Nameserver Co-location

As we have discussed in Section 4.4, there are some "hot spots" of nameserver placement at different granularities. A lot of server nodes of different domains are co-located at these "hot spots". Moreover, if we revisit Figure 6 and 7, it is easy to observe a high dense area at the left bottom corner in each figures, especially in Figure 6. These areas indicate that when operators of TLDs decide where to deploy redundant server nodes, they may have the preferences on the same set of server provider, or they may prefer those already hosting a large amount of server nodes. In rest part of this section, we investigate the existence of co-location, and evaluate its impact on the robustness of TLD services.

6.2.1 Existence of Nameserver Co-location

We investigate the existence of nameserver co-location at four granularities: 1) Server provider, 2) AS, 3) City, and 4) Country. At each granularity, a point is hotter than others if there are more server nodes at that point. The only exception is server provider. Since all server nodes of a given nameserver belong to the same server provider, it is more reasonable to use the number of nameservers, rather than the number of server nodes as its co-location metric.

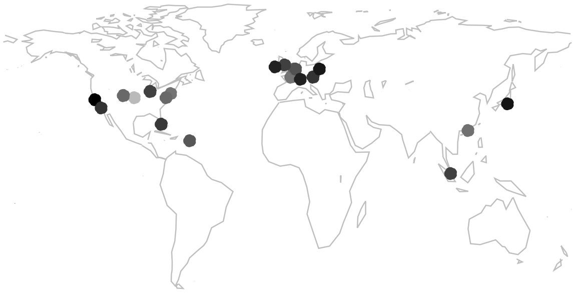


Figure 12: Hot spots of TLD services on cities.

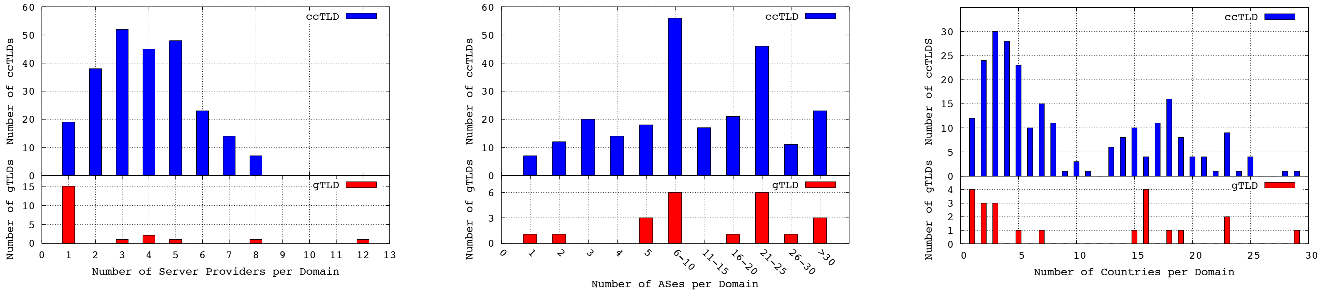
The nameserver co-location is shown in Figure 11. We can observe obvious hot spots of different granularites in current TLDs. In Figure 11a, although 80% ccTLD nameservers are provided by 208 server providers (50% of total ccTLD nameserver providers), top 10 providers (2%) control 35% ccTLD nameservers. 80% gTLD nameservers are co-located at 7 server providers (25% of total gTLD nameserver providers). The hottest points are large commercial DNS service providers and non-profit network service providers, such as VeriSign, NeuStar, RIPE and etc..

In Figure 11b, 80% server nodes of ccTLD nameservers are co-located at 76 ASes (14% of total ASes hosting server nodes of ccTLDs), and top 10 ASes (2%) host 37% server nodes of ccTLDs. 80% server nodes of gTLD nameservers are co-located at 34 ASes (30% of total ASes hosting server nodes of gTLD). The hottest points are ASes of Tier-1 ISPs, such as NTT, Cogent, Level 3 and etc..

In Figure 11c, 80% server nodes of ccTLD nameservers are co-located at 19 countries (11% of total countries hosting server node of ccTLDs), and top 10 countries (4%) host 60% server nodes of ccTLDs. 80% server nodes of gTLD nameservers are co-located at 10 countries (25% of total countries hosting server node of gTLDs). The hottest points are countries with well deployed internet infrustructures, such as U.S., Netherland, U.K. and etc..

Figure 12 shows the name server co-location of DNS services on the top 20 cities. It should be pointed out that, the accuracy of Figure 12 is depended on Maxmind's IP location inference database. On this map, each circle indicates a hot spot at the granularity of cities. The grey degree of a circle represent the number of TLD server nodes in a city. The more server nodes a city hosts, the darker the cirle is. 57.8% of TLD server nodes are placed in the top 20 cities (5% of total cities hosting TLD server nodes). The hottest points, such Palo Alto (U.S.), Amsterdam (Netherlands), London (U.K.) and etc., are rich in network connectivities.

These hottest points are either rich in network connectivities, or experienced in DNS service provision-

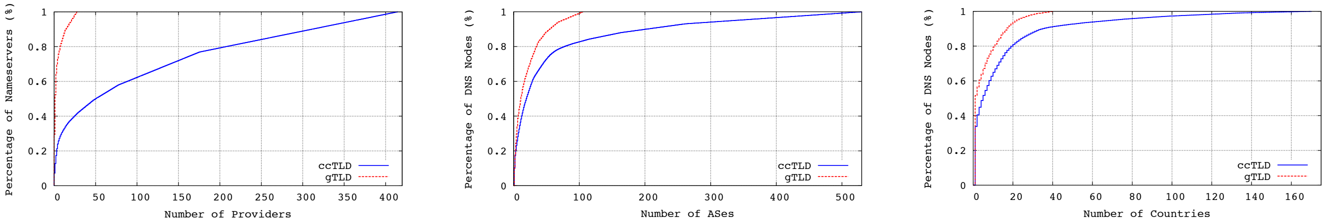


(a)

(b)

(c)

Figure 10: Location diversity of TLDs at different granularities: (a) server providers; (b) ASes; (c) countries



(a)

(b)

(c)

Figure 11: CDF of nameserver co-location on: (a) Server providers, (b) ASes and (c) countries.

ing, or large commercial DNS service providers. They confirm our discussion about causes of nameserver co-locations in Section 4.4.

6.2.2 *Impact of Nameserver Co-location*

To evaluate the impact of Nameserver co-location on the robustness of TLD services, we conduct following experiments at each granularity: We select hot spots in a specific order, and simulate failure of the selected hot spot by removing all server nodes co-located at the spot. Then we count the number of ccTLDs and gTLDs surviving after the removal, i.e. still having available server nodes. We continue to select another hot spot from the rest, and repeat the same procedure. Spots are selected in two sequences: descending (i.e. starting from the one with the most server nodes) and ascending (i.e. starting from the one with the fewest server nodes). Figure 13 shows the simulation results on ASes, providers, countries, and city-level locations.

Experiment 1 (represented by the green line) used the descending sequence. Experiment 2 (represented by the orange dashed line) used the ascending sequence. As we can see in Figure 13 (Country), the green line starts to drop after first 10 countries has been removed. This indicates that, although the top 10 hot spot countries possess an overwhelming portion of total server nodes (60%), their failures has little impact on the availability of ccTLD's DNS services. We can also observe that the green line drops linearly after 40 countries have been removed. It should be noted that rest countries are "small countries", i.e. they have a small number of server nodes in their countries, in most cases only one server node for its own ccTLD. The linearly droping green line indicates that ccTLDs of these small countries do not depend on these large countries, i.e. hot spot countries. Even if those hot spot countries fail, ccTLD services of these small countries will keep working until server nodes in their own countries fail.

Experiment 2 revealed another effect of nameserver co-locations. The orange line drops very slowly when "small countries" are being removed, but drops dramatically at tail when removing "large countries". This indicates that most ccTLDs place at least one server node in those hot spot countries. Therefore, as long as network accessibilities of those large countries are kept in a good condition, DNS service of most ccTLD can still be available, no matter what has happened in their own countries. Due to rich network connectivities and robust internet infrastures, it is more difficult to break down networks of those "large countries" (e.g. U.S., Netherlands) than those small countries. Therefore, it would be desired to keep those hot spots as long as each ccTLD can guarantee at least one server node in its own country.

We also conduct the same experiments on city-level locations, ASes, and server providers, and got similar

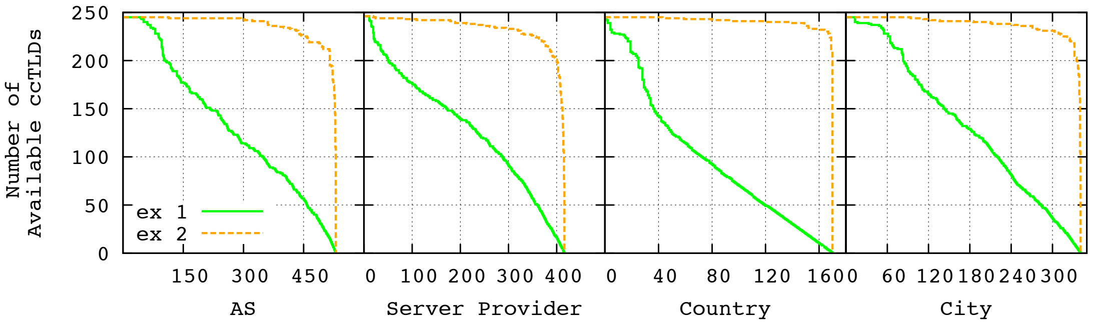


Figure 13: Simulations of hot spot failure's impact on ccTLD's

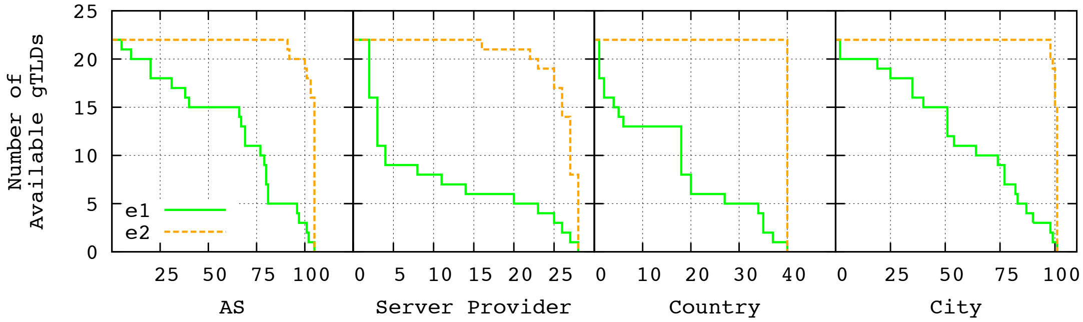


Figure 14: Simulation of hot spot failure's impact on gTLD's

results as shown in Figure 13. DNS services of all ccTLDs keep available until top 50 ASes have failed. The tail of descending sequence also drops linearly, as many countries host their server nodes in their own ISPs' ASes. And we can also observe that the top 15% ASes can guarantee DNS service availability of more than 95% ccTLDs. For server providers, DNS services of all ccTLDs can still be accessed until top 10 server providers have failed. The tail of descending sequence drop linearly, as many countries place at least one of their server nodes in their local server providers. Similarly, the top 13% server providers can guarantee 90 % ccTLDs' DNS service availability. As to city-level locations, all ccTLDs can survive from failures of top 12 cities. And top 10% cities can guarantee DNS service availability of more than 95% ccTLDs.

gTLDs exhibit almost the same results as shown in Figure 14. The small difference is caused by the fact that most gTLDs usually place all their nameservers at one server provider. As we can see in Figure 14, 7 gTLDs maintained by VeriSign will fail if VeriSign fails. However, due to widely deployed anycast on gTLDs' nameserver, concentration on cities, countries, and ASes has less impact on gTLD's service availablity than server providers.

From analysis above, we can conclude that nameserver co-location exists in current TLD nameserver deployment. A huge amount of server nodes are located at a small number of hot spot, so that they can cover almost all the ccTLD DNS services. But these hot spots affect the robustness of ccTLDs little, because most of ccTLDs have at least one server node outside these hot spots. Additionally, hot spots overlap with each other on the ccTLDs they support, no ccTLD will fail unless all of these hottest points fail at the same time. For gTLDs, since they are usually maintained by one server provider, their robustness is mainly affected by the co-location at server providers. But they are less affected by geographical and topological co-location, because impacts of these co-locations have been mitigated by anycast.

7. CONCLUSION

In this work, we conducted a systematic measurement study on robustness of TLD services regarding the redundancy and location diversity of their nameserver placement. We measured locations of all TLD nameserver, and identified their providers. We examined the location diversity of both gTLDs and ccTLDs. We found that gTLDs had good diversity on topographical and geographical locations due to widely deployed anycast nameservers. However, since most gTLDs delegated all its nameservers to only one server provider, more than 70% gTLDs may suffer from single point failures on server providers. Most of ccTLDs also had good location diversity. The dominant reason is that they delegate some of their nameserver to server providers supporting anycast servers. However, the nameserver delegation leads to another placement issue of ccTLDs. We found that more than 30% ccTLDs do not host nameservers inside their own countries. These ccTLDs will become unavailable to people in these countries, if these countries lose network connectivities to the outside world.

We also observed that a large amount of server nodes belonging to different TLDs are co-located at a small number of "hot spots". Such hot spots exist at different granularities, such as countries, cities, ASes and server providers. We attribute the existence of co-location to three reasons: rich network connectivities, operators' preference and economic considerations. We also evaluated the impact of hot spots failure on robustness of TLD services. We found that the impact is little because most TLDs, except gTLD with only one server provider, put at least one server node outside these hot spots.

8. REFERENCES

[1] EyeP, http://eyep.cs.ucla.edu.

[2] Maxmind, http://www.maxmind.com.

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

[4] rdnsD - Reflected DNS Daemon, http://www.netsec.colostate.edu/dnsmonitor.

[5] Root Server Technical Operations Assn, http://www.root-servers.org.

[6] UCLA IRL Internet topology collection, http://irl.cs.ucla.edu/topology/.

[7] J. Abley. Hierarchical anycast for global service distribution, 2003.

[8] J. Abley. A software approach to distributing requests for DNS service using GNU Zebra, ISC BIND 9 FreeBSD. In *Proceedings of the annual conference on USENIX Annual Technical Conference*, pages 18–18. USENIX Association, 2004.

[9] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira. Avoiding traceroute anomalies with paris traceroute. In *Internet Measurement Conference*, pages 153–158, 2006.

[10] R. Austein. RFC 5001, DNS Name Server Identifier (NSID) Option. 2007.

[11] H. Ballani, P. Francis, and S. Ratnasamy. A measurement-based deployment proposal for IP anycast. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 231–244. ACM, 2006.

[12] N. Brownlee, K. Claffy, and E. Nemeth. DNS Root/gTLD performance measurements. *USENIX LISA, San Diego, CA*, 2001.

[13] L. Colitti, E. Romijn, H. Uijterwaal, and A. Robachevsky. Evaluating the effects of anycast on DNS root name servers. *RIPE document RIPE-393*, 2006.

[14] R. Elz, R. Bush, S. Bradner, and M. Patton. RFC2182: Selection and Operation of Secondary DNS Servers. 1997.

[15] S. Gibbard and P. House. Geographic implications of DNS infrastructure distribution. *The Internet Protocol Journal*, 10(1):12–24, 2007.

[16] T. Hardie. RFC 3258, Distributing Authoritative Name Servers via Shared Unicast Addresses. 2002.

[17] T. Lee, B. Huffaker, M. Fomenkov, et al. On the problem of optimization of DNS root servers' placement. *Passive and Active Network Measurement Workshop (PAM)*, 2003.

[18] R. Liston, S. Srinivasan, and E. Zegura. Diversity in DNS performance measures. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 19–31. ACM, 2002.

[19] M. Luckie. Scamper: a Scalable and Extensible Packet Prober for Active Measurement of the Internet. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 239–245. ACM, 2010.

[20] D. McPherson. Unique Per-Node Origin ASNs for Globally Anycasted Services, http://tools.ietf.org/html/draft-ietf-grow-unique-origin-as-00.

[21] P. Mockapetris. RFC 1034, Domain Names-Concepts and Facilities. 1987.

[22] V. Pappas, Z. Xu, S. Lu, D. Massey, A. Terzis, and L. Zhang. Impact of configuration errors on DNS robustness. In *ACM SIGCOMM Computer Communication Review*, volume 34, pages 319–330. ACM, 2004.

[23] S. Sarat, V. Pappas, and A. Terzis. On the use of anycast in DNS. In *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 394–395. ACM, 2005.

[24] S. Woolf. RFC 4892, Requirements for a Mechanism Identifying a Name Server Instance. 2007.

[25] Y. Zhang, R. Oliveira, H. Zhang, and L. Zhang. Quantifying the pitfalls of traceroute in as connectivity inference.