On the Use of Anycast in DNS

Sandeep Sarat
sarat@cs.jhu.edu
Johns Hopkins University

Vasileios Pappas
vpappas@cs.ucla.edu
UCLA

Andreas Terzis
terzis@cs.jhu.edu
Johns Hopkins University

Abstract

In this paper, we measure the performance impact of anycast on DNS. We study four top-level DNS servers to evaluate how anycast improves DNS service and compare different anycast configurations. Increased availability is one of the supposed advantages of anycast and we found that indeed the number of observed outages was smaller for anycast, suggesting that it provides a mostly stable service. On the other hand, outages can last up to multiple minutes, mainly due to slow BGP convergence. We also found that anycast indeed reduces query latency. Furthermore, depending on the anycast configuration used, 37% to 80% of the queries are directed to the closest anycast instance. Our measurements revealed an inherent trade-off between increasing the percentage of queries answered by the closest server and the stability of the DNS zone, measured by the number of query failures and server switches. We believe that these findings will help network providers to deploy anycast more effectively in the future.

***Index Terms*–Anycast, Routing, Internet, DNS, Root Servers.**

1 Introduction

Anycast is widely deployed in DNS today [11]. The IP addresses of many top level DNS nameservers correspond to anycast groups. Anycast provides a service, whereby a host transmits a datagram to an anycast address and the internetwork is responsible for delivering the datagram to at least one, preferably the closest, of the servers in the anycast group [17]. Operators can incorporate anycast in their DNS zones in a number of ways. For example, they can use single or multiple nameservers addresses each with a different anycast address. Anycast prefixes can be globally advertised or their scope can be limited to the immediate neighboring autonomous systems. Servers whose advertisements are scoped are called *local nodes* while nodes with no scoping are called *global nodes*. Local nodes limit the visibility of their advertisements by using the `no-export` BGP attribute. Peers receiving advertisements with this attribute should not forward the advertisement to their peers or providers. Scoping is used to support servers with limited transaction and bandwidth resources and servers serving only local networks. Finally, the anycast prefix(es) can originate from a single AS or the zone operator can be *multi-homed* so multiple ASes inject the prefix to the global BGP table.

Anycast is believed to be possess a variety of advantages: reduced query latency, increased reliability and availability as well as resiliency to DDoS attacks. This paper presents a comprehensive survey of anycast deploment, thus evaluating the performance benefits of deploying anycast. Specifically, we aim to answer the following questions: (1) Do servers deploying anycast experience smaller number of outages (2) How stable is the anycast server selection over time? (3) Does anycast reduce query latencies? To answer these questions, we performed a measurement study using clients deployed over PlanetLab [19], to measure the performance characteristics of four top-level servers using anycast and compared it to a server not using anycast. We monitored servers that represent different points on the anycast design space. Specifically, we evaluate the effects of single vs. multiple anycast addresses for a zone and global vs. localized visibility of the servers in the anycast group. We also compared these servers against a hypothetical zone with the same number of nameservers but where all the nameservers are individually addressable. By doing so, we can directly compare anycast to the traditional zone configuration guidelines [7].

We found that for all the measured zones, the deployment of anycast decreases average query latency and increases availability when compared to centralized servers. Furthermore, our study shows that while the number of query failures is relatively small ($\leq 0.7\%$), outages are long in duration ($\approx 30\%$ last more than 100 seconds), affected by long BGP routing convergence times. Interestingly, we show that, even though the outage duration is not affected by the anycast scheme, the frequency of the outages relates to the scheme used, i.e. whether servers have local or global visibility. In addition we identified that the anycast scheme determines the percentage of queries directed to the clos-

est anycast instance. This value ranges from about 37% for servers with a few global nodes to about 80% for servers, wherein all nodes are global. We also uncovered an inherent trade-off between increasing the effectiveness of anycast in directing the queries to the nearest server and stability of the zone itself. For servers that advertise all their anycast group members globally, clients choose the nearest server most of the time. The negative effect though is that, in this case the zone becomes vulnerable to increased number of network outages and server switches.

The rest of the paper is structured as follows: We explain our measurement methodology in Section 2. We present our results and compare the different anycast strategies in Section 3. Finally, we present related work in Section 4 conclude in Section 5.

2 Measurement Methodology

Our goal is to investigate the implications of using anycast in DNS and to compare the performance benefits of different anycast configurations. The two primary factors affecting the performance of anycast are: **(I)** the number and location of the anycast servers relative to the DNS clients, and **(II)** the anycast scheme used. To quantify the relative benefits of each of these factors we used four types of server configuration in our measurements, each representative of a different point in the anycast design space.

We used the PlanetLab [19] testbed for our measurements. We collected data from the PlanetLab nodes from September 19, 2004 to October 8th, 2004. At the time of our measurements, there were approximately 400 nodes in PlanetLab contributed by universities and research labs around the globe. The results presented in this paper are based on measurements from approximately 300 active PlanetLab nodes. As we already mentioned, the client locations relative to the servers can potentially affect our measurements. North America had $\sim$65%, Europe and Asia $\sim$17% each and the rest in South America, Australia and Africa.

We ran a script on every PlanetLab node to send periodic DNS queries to each of the DNS servers earlier mentioned. The query interval is selected at random from [25,35] seconds. We recoreded the query latency and the server name corresponding to the anycast instance answering the query. The scripts uses "special" DNS requests to retrieve the name of the server replying to a request sent to the anycast address ([2] shows the request type for F-root).

2.1 Anycast Deployments

The configuration of the monitored anycast servers in the anycasy design space are as follows:-

Multiple Instances, One site. e.g B-Root The B-Root server has 3 nodes (b1/b2/b3.isi.edu), and all of them reside in the same network (located in Los Angeles, CA).

Multiple Instances, Multiple Heterogeneous Sites. e.g F,K-root This is the case where an anycast server has multiple instances deployed in geographically diverse sites, with some instances being globally visible and the rest, being scoped in their local region. At the time of the experiment, F-Root had 26 sites out of which two were global and the rest local. The global nodes PAO1 (Palo Alto) and SFO2 (San Francisco) served $\sim$39% and $\sim$32% of the planetlab requests respectively. We were able to contact a total of 19 instances using planetlab.

K-root, which is similar to F-Root, is primarily concentrated in Europe, but has lesser anycast group members[13]. Clusters at Amsterdam(AMS-IX) and London(LINX) have global visibility, while the rest local visibility About ($\sim 97\%$) of PlanetLab nodes are served by LINX and AMS-IX. Using planetlab, we were able to contact a total of 4 instances out of a group size of 7.

The limited visibility and the skewed distribution of requests served by the group members is because in either of the above configurations some nodes are deployed as global nodes [1]. The rest of the clusters are visible locally and serve clients only within their communities. Since the unreachable nodes have local scope, no PlanetLab nodes are located within their scope, because the AS path to the global node is shorter than the path to the local node. Therefore the PlanetLab site chooses to route requests to the global node instead of the local.

Multiple Instances, Multiple Homogeneous Sites. e.g UltraDNS We used the two UltraDNS anycast servers (TLD1 and TLD2) as representative cases of anycast servers having multiple instances in diverse geographics location, all of them globally visible. The benefit of using two anycast servers is that in the event of a network outage affecting one of the anycast addresses, the other address can be used, ensuring uninterrupted DNS service. Due to the unavailability of the complete listing of UltraDNS clusters, we only consider clusters that are reachable from PlanetLab nodes. Unlike F- and K-Root, UltraDNS uses a flat setup, where BGP advertisements from all instances are globally visible throughout the Internet. Thus, DNS requests are more evenly distributed across UltraDNS clusters, compared to the F-, K-root servers. Instances in Europe (`abld`) and Asia (`eqhk`) serve a smaller percentage of nodes since fewer PlanetLab nodes are located in these continents. These two nodes do not serve TLD2 requests. The distribution of client requests across TLD1 and TLD2 is also totally different. For example, while `pxpa`(Palo Alto,CA) receives 23% of the queries for TLD1 it receives only 7% of the queries for TLD2. The reason why PlanetLab nodes mostly pick different clusters for TLD1 and TLD2 name resolution, is that UltraDNS uses two different carriers for TLD1 and TLD2 BGP advertisements. This fact was corroborated us-

ing data from Routeviews [20] and traceroutes from PlanetLab nodes to tld1/tld2.ultradns.net.

Multiple Instances accessible via unicast A set of geographically distributed servers, each individually accessible via unicast. We used this case to evaluate the quality of the routing paths provided by the network fabric connecting the anycast servers to their clients. To enable a direct comparison with anycast, we want to keep the number and location of the name servers constant. To do so, we used the F-root example, but in this case clients send requests to the unicast addresses of the F-root group members. Each client maintains an ordered list of all the servers based on their latency and sends its queries to the closest server from its list. In case a server becomes unavailable, the client tries the subsequent servers in its list until it receives a response.

3 Evaluation

This section examines: (1) the query latencies for the monitored servers; (2) the availability of the monitored servers; (3) the affinity of clients to the server they are directed to and (4) the percentage of clients not reaching the replica server that is closest to them and the additional delay incurred.

3.1 Response times

Table 1 presents the mean, median, and standard deviation of query latencies for the monitored servers over the whole measurement period. The median provides a better indication of the expected behavior, since it is not skewed by individual clients with very high latencies. The first observation we can make from this table is that anycast provides a sizable reduction in query latency compared to the B-root server. The only exception to this trend is K-root. This is due to the fact that even though K-root has multiple servers, they are located in Europe and the Middle East, while most of the PlanetLab nodes are in North America. Second, TLD1 has the lowest latency, even though F-root has more deployed servers. The reason is that only two of the F-root servers have *global* scope and therefore client requests may have to travel to a server that is further away. On the other hand, UltraDNS does not use scoping and client requests are distributed among a larger set of geographically diverse servers leading to shorter round trip times. Furthermore, the median latency for TLD1 is lower than that of TLD2 since clusters `abld` and `eqhk` are not reachable for the TLD2 anycast address. Therefore queries to TLD2 from clients in Europe and Asia have to travel to the US.

The last two rows of Table 1 represent synthetic results derived from actual measurements. The min{TLD1,TLD2} row represents the average query latency for clients that choose the closest server between TLD1 and TLD2, to direct their queries to. Remember that UltraDNS, which is authoritative for the .org and .info top level domains, uses two anycast addresses for these domains' nameservers. So this row represents the best case scenario where a client can measure the latency to each of the nameservers and subsequently direct its queries to the closest nameserver. Indeed, clients based on BIND 9 exhibit this behavior [23]. The last row of Table 1 shows the average latency of the hypothetical zone where all the F-root servers are directly accessible by their unicast addresses and clients forward their request towards the closest DNS server. The latency of this zone is lower than F-root due to *scoping*. As we already mentioned, scoping leads clients to pick a server that is further away since the announcements from servers with local scope that are closer than the global server do not reach them.

Nameserver	**Mean** (ms)	**Median** (ms)	**Std. Dev.** (ms)
F-Root	75	70	85
B-Root	115	95	121
K-Root	140	121	104
TLD1	96	54	207
TLD2	104	85	237
min{TLD1,TLD2}	69	51	173
Hypothetical unicast	45	35	13

Table 1. Statistics of DNS response times

TLD1 and TLD2 exhibit the highest variance in the response times across all measured servers. This is due to two reasons: variability in the delay of the network paths and variability in the load on the anycast server. As we already explained, UltraDNS anycast addresses are globally announced. In Section 3.3 we show that this results in clients experiencing a higher number of "flips" (i.e. server changes), and consequently higher fluctuation in DNS response times.

3.2 Availability

Considering the reliance of most Internet applications on DNS, ensuring continued availability is a prime requirement for top level name servers. We retry individual unanswered queries twice and therefore the results presented here indicate queries lost due to network and server outages rather than random packet loss. For all the measured servers, the average percentage of unanswered queries is low ($\leq 0.9\%$). (B-Root - 0.87% ,F-Root - 0.35%, K-Root - 0.5% , TLD1 - 0.72%, TLD2 - 0.6%). At the same time, the benefit of deploying servers in multiple locations is evident from the fact that all anycast schemes perform better than B-Root. This is to be expected since robustness generally increases with geographic diversity. This is the reason why F-Root has

smaller percentage of unanswered queries compared to K-Root even-though both of these servers use the same anycast scheme. There is however large variation between the availability of the different anycast schemes, with F-root having overall half the losses of TLD1.

We use the term "outage" to indicate a window of time when a node is unsuccessful in contacting its DNS server. Figure 1 plots the CDF of the duration of outages for the different servers. The first observation from the graph is that all outages last at least 20 seconds because of the time granularity with which we send DNS requests. Second, outages for the hypothetical unicast server have the shortest duration. Indeed some (20%-30%) of the PlanetLab experienced no outages. The maximum outage time is around 100 seconds, indicating that in the worst case, a client will get a response after contacting at most three servers. At the same time, the mean outage duration is approximately 40 sec, two to three times shorter from the other servers. The min{TLD1,TLD2} combined nameserver enjoys the same benefit of shorter outage periods since clients can switch from a failed server in one of the anycast addresses to a server in the other address. All the real world anycast deployments exhibit similar distribution in outage time with F-root having the longest outage periods. This reveals an interesting fact regarding anycast. Since anycast relies on Internet routing, once an outage has occurred the recovery time is governed by the recovery time of the network routing fabric. In fact, $\approx 30\%$ of the outages last more than 100 seconds. This is a direct consequence of the results presented by Labovitz et.al regarding delayed network convergence [14]. The outage recovery time is largely independent of the anycast scheme used.

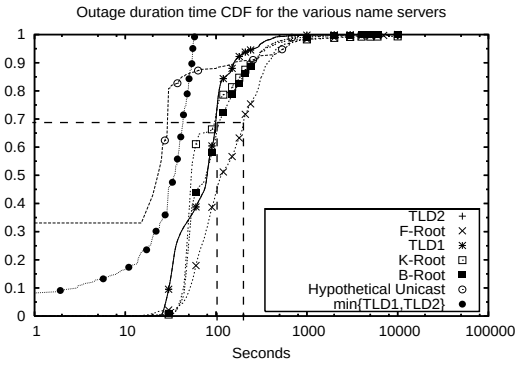

Figure 1. CDF of outage duration

It appears counter-intuitive that F-root can have the smallest percentage of lost queries and at the same time have the longest duration outages. However, outage duration is only one part of the picture. It is also important to note *inter-outage* interval and the number of outages which occur per server. Figure 2 shows that inter-outage intervals, that is the amount of time between succesive outages experienced by the same client. The findings from this graph are encouraging as they show average inter-outage periods in the order of days. At the same time, inter-outage periods for TLD1 and TLD2 are shorter than those for F-root. Supporting this fact, we observed see that TLD1 and TLD2 experience five to eight times more outages than F-root. The reason why TLD1 and TLD2 have higher percentage of unanswered queries, even though the duration of their outages is shorter is that outages occur more frequently giving a larger total number of unanswered queries.

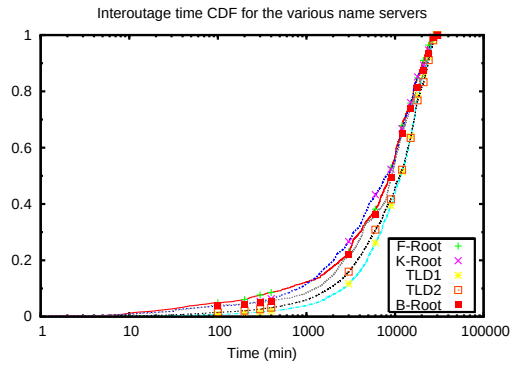

Figure 2. CDF of inter-outage duration

While at this point we don't fully understand why UltraDNS experiences more outages than F-root, we conjecture that this is due to two reasons: First, all UltraDNS clusters are global. As a result, clients follow more different paths to reach their servers and are therefore more exposed to BGP dynamics when links fail. Second, TLD1 and TLD2 are single-homed while F-root is multi-homed. As a result if the first-hop ISP of TLD1 fails, all TLD1 clusters become unavailable. On the other hand, since F-root is multi-homed the impact of any single ISP failure on the overall availability is smaller.

3.3 Constancy

Since anycast groups consist of multiple nodes, destinations will change over time as routing adapts to network changes. In this section we present our findings on server switches (or flips) for the monitored anycast servers. We classify flips into two categories: *inter-cluster* and *intra-cluster*. An inter-cluster flip happens when consecutive client requests are directed to two different geographic clusters and is due to BGP changes. Each of these clusters contains multiple DNS servers and an intra-cluster flip happens when the same client is directed to different members located inside the same cluster. Inter-cluster flips are due to local load

balancing at the anycast cluster. As we saw in Sec. 3.1, the rate of flips affects the query latency variance. Delay consistency is more sensitive to inter-cluster flips than intra-cluster ones, because inter-cluster flips involve a change of transit route, and different routes may have widely different delay characteristics.

Inter cluster flips in anycast deployments using global and local servers mostly occur between the global servers. The majority of the flips ($> 90\%$) for F-Root are between the PAO and SFO global clusters, while for K-Root between AMS and LINX. Furthermore the total number of inter-cluster flips observed in the F-Root and K-Root nameservers is 20% lower compared to TLD1 and TLD2. We believe the reason for this is that UltraDNS anycast clusters are globally visible while the majority of F-Root and K-Root are local clusters. Therefore at a client gateway, BGP paths to a greater number of UltraDNS clusters are available compared to F-root clusters. Hence, UltraDNS server selection is more prone to BGP changes (due to path failures).

Nameserver	Flips (%) of queries	Flips linked to an outage (%)
F-Root	0.0055	65
K-Root	0.0060	63
TLD1	0.0072	52
TLD2	0.0074	51

Table 2. Percentage of flips due to outages.

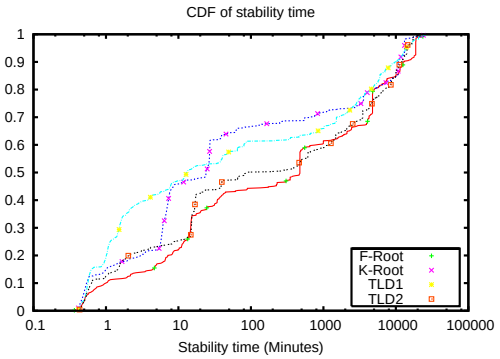

Figure 3. Period of time that PlanetLab nodes query the same server for the monitored servers.

Flips and outages can often be correlated. For example, on Sep. 21st, a considerable number of PlanetLab nodes faced outages in the service from the SFO cluster of the F-Root server. After a brief outage of over a minute, service resumed with nodes contacting the PAO cluster for F-root name resolution instead. Similarly, on Sep. 27th for the K-Root server, all PlanetLab nodes using the AMS cluster experienced an outage. After a brief interval spanning over two minutes, all these nodes flipped to the LINX cluster. However, flips need not necessarily occur immediately after outages. To investigate how strongly flips are correlated to server outages, we counted the number of flips that are *linked* to an outage. When a client flips to a different server after the server it was using becomes unavailable and later flips back to the original server, we say that these two flips are related to the server outage. As Table 2 shows, in the case of TLD1 and TLD2 UltraDNS servers, the occurrence of flips and outages are related to a lesser extent. Since UltraDNS clusters are all global nodes, flips are more frequent and half of the time occur independently of outages. We believe two causes are behind the remaining flips: path changes in Internet routing and path failures recovered by the routing infrastructure within the inter-query interval (25-35 seconds).

The percentage of flips across all the servers is very small, indicating that they offer a stable service. We are also interested in the time that PlanetLab nodes remain stable to the same server. We found that there is a range of 5 orders of magnitude in this metric! As Figure 3 illustrates, while the mean time a node remains stable to the same server is around 100 minutes, the lowest 10% of the nodes change servers every 1 minute, while the most stable clients consistently choose the same server for days or weeks. This behavior is evidence that a small number of network paths are very stable while most other paths suffer from outages and a small percentage of paths have a pathological number of outages. Furthermore, for servers that use global and local nodes (i.e. F- and K-root), we found that global nodes are more prone to switches as we already mentioned. We believe the reason for this behavior is that the network paths to global nodes are longer and therefore more prone to BGP dynamics.

Intra-cluster switches: Load balancing also occurs inside clusters, to distribute queries among the individual servers that make up the cluster. F-root uses IGP-based (OSPF) anycast for load balancing [2], but other configurations could use hardware based load balancers. Load balancers use either a per-packet or a per-flow mechanism. To discover the load balancing scheme in the nameservers, we use to our advantage the fact that each PlanetLab sites contains multiple nodes. These nodes can be expected to contact the same anycast cluster. The similarity between the anycast servers of single site nodes provides a hint to the type of load balancer used within each cluster. Large correlation between the servers contacted by the nodes of the same site, indicates a per-packet load balancer (given a round-robin load-balancing scheme we expect that packets from each client will be sent to all the servers inside the DNS cluster). On the other hand, low correlation indicates flow

based load distribution (a technique often used by load balancer hashes the clients' source addresses). Using this technique we discovered that all the candidate nameservers used a flow based technique, except for the B-Root server which used a per packet load balancer. We observed that the B-root server faced a flip every half a minute. This is typical of a per-packet load balancing technique, where successive data packets are sent to different servers without regard for individual hosts or user sessions.

3.4 Effectiveness of Localization

As our earlier results indicate, anycast decreases query latencies by localizing client requests amongst the various DNS server replicas. However, comparing the F-root query latency to that of the hypothetical zone where all the servers are individually addressable (Table 1) seems to suggest that anycast does not always pick the closest server. This raises the interesting question: Does anycast always lead clients to the closest instance among all the servers in the anycast group? If not, how much farther away is the selected server as compared to the closest? Anycast server selection depends on the path selected by BGP. These routing decisions are influenced by policies and sub-optimal heuristics such as using the path with the shortest AS hop count and can therefore lead to suboptimal choices. In fact, it is well known that in many cases the paths chosen by BGP are not the shortest [21, 22].

An Optimistic Estimate: Directly comparing the query times of requests sent to the unicast addresses of all the anycast group members, to the query time of the requests sent to the server selected by anycast is potentially flawed due to a subtle reason. Typically, the unicast addresses of the DNS servers are selected from address ranges that are different from the one used for anycast, for reasons of ease of management and robustness. Therefore the path from a client to the anycast address can be different from the path to the unicast address of the same server.

We use the following technique to get around this difficulty. Our technique is based on the fact that if traceroutes from a client to the last hop router and the anycast address follow the same path, we can obtain a good approximation of the round trip times incurred by a client query to each of the different clusters by using the round trip time to the last hop router instead. Using traceroutes from the PlanetLab nodes, we found that this was indeed the case for the F-Root and TLD2 servers, but not so for TLD1 and the K-Root. Figure 4 presents the additional network latency incurred by clients following the path to the server selected by anycast over the path to the closest server. One can see that in both cases the majority of the anycast queries contact their nearest cluster. About 60% of all the F-Root requests are sent to the nearest F-cluster and 80% of the TLD2 requests are sent to the nearest TLD2 cluster. It must be however be noted that this is an upper bound on the optimality of the anycast path choice for F-root, as not all the anycast clusters are visible to the PlanetLab nodes.

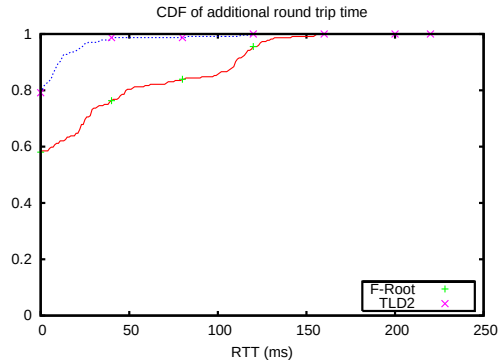

Figure 4. Additional round trip time for client queries to the anycast-selected F-root and TLD2 servers over the closest servers.

A Pessimistic estimate: We also measured the effectiveness of localization using another approach, which yields a lower bound on the effectiveness of localization. First, we calculate the geographic distance of each of the PlanetLab nodes to all the listed DNS clusters in a zone. We do so by calculating the length of a hypothetical straight line over the globe connecting the geographic locations of the PlanetLab node and the DNS server. The locations of PlanetLab nodes are available through the PlanetLab website. Then, we compare these geographic distances and determine whether the PlanetLab node contacts the geographically closest server in that zone. While it is known that Internet paths are longer than the direct geographic path connecting two end-points [9, 22], we assume that all paths exhibit the same *path inflation* factor. Based on this assumption, we can directly compare geographic distances to determine whether the best Internet path is selected for each client.

Figure 5 presents the cumulative distribution of the additional distance across all PlanetLab nodes for each zone. We observe that about 37% of all the anycast requests are sent to the nearest F-root server while 35% of the anycast requests are sent to the nearest K-root server. Approximately 75% requests are served by the nearest TLD1 and TLD2 servers. In fact, the CDF for TLD2 closely matches with that in Figure 4. However, this is not the case with F-Root, because not all the clusters are visible from Planet Lab and consequently not accounted for in Figure 4.

Using these two estimates, we can conclude that the effectiveness of localization for the F-Root is between 37%-

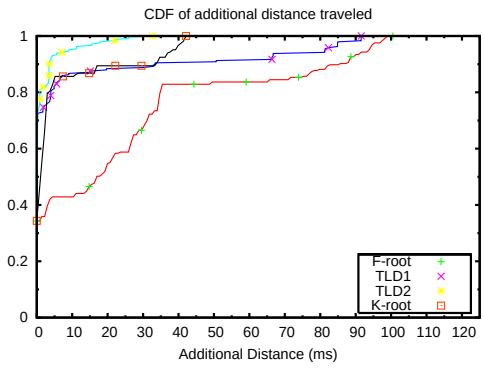

Figure 5. Additional distance over the optimal traveled by anycast queries to contact their F-root, K-root, TLD1 and TLD2 server.

60%, $\geq$ 35% for the K-Root, while for TLD1, it is $\geq$ 75% and for TLD2 between 75%-80%. It is not surprising that TLD1 and TLD2 zones perform considerably better than other deployments. Not only a larger portion of nodes contact the closest server but the additional distances for those that don't, are also shorter. The reason is that UltraDNS clusters are not differentiated into global and local. Consequently, PlanetLab nodes have visibility to a greater number of BGP routes to UltraDNS clusters. Therefore, it is more likely that anycast chooses the nearest UltraDNS cluster. In a somewhat counter-intuitive way, the slowest 10% of TLD1 clients follow worse paths compared to TLD2 even-though TLD1 is advertised by two additional locates (London, Tokyo). We explain this behavior by an example. Consider a client in Asia. If it doesn't pick the HK site for TLD1, its requests are directed to the US. Thus the large additional distance. On the other hand, TLD2 is not advertised locally from HK and therefore clients correctly pick the US sites. The inverse effect is visible for K-Root. Clients don't traverse large additional distances compared to the closest cluster due to the fact that all their clusters are located within a relatively small geographical area.

3.5 Comparison of Deployment Strategies

We can categorize the existing anycast configurations into two schemes: hierarchical and flat. The hierarchical scheme distinguishes anycast nodes into local and global, while in the flat scheme all the nodes are globally visible. Anycast servers in the flat configuration tend to have a more uniform distribution of load. Also, since there is a greater diversity of choices of available anycast servers to a client, the distance between clients and DNS servers is generally shorter, as seen in Section 3.4. Consequently, majority of the clients also have low query latency as reflected in the low median query time of TLD1 and TLD2 anycast servers, in Section 3.1. However, in Section 3.2 we see that the flat scheme is more prone to outages. Even though the outage durations follow similar distribution for both schemes, given that it is a function of the BGP convergence time, the frequency of the outages is lower for the hierarchical scheme. That is possibly due to the fact that in the case of the flat scheme more instances are globally visible in the routing tables, and thus they can potentially lead to path changes triggered by other network events. Furthermore, in Section 3.3 we show that having a large radius of advertisements has an adverse effect on the stability of the response times and increases the frequency of server changes (flips) of the anycast service. This is because the larger the radius of advertisements is, the greater is a server's sphere of influence. This consequently increases the number of choices of servers available at a client.

We believe that an ideal anycast scheme would involve deploying a small number of global nodes accompanied by a larger group of local nodes. The radius of advertisement of the local nodes can be dynamically varied in order to maintain a minimum degree redundancy and fast failover.

4 Related Work

A number of existing studies have looked at the performance of the DNS infrastructure. Brownlee *et al.* monitored the DNS traffic from a large campus network and measured the latency and loss rate of queries sent to the root nameservers [6]. Their main goal was to create a model of DNS request/response distribution. Our results on average latencies and loss rates match those presented in that study. Interestingly, the authors of [6] observed that query times show clear evidence of multipathing behavior and conjectured that this is due to load balancing or changes in server load. Anycast at the BGP level and within a cluster, is a key cause of this observed multipathing. Pang *et al.* [15] measured the availability of the individual DNS authoritative and caching servers, and studied the different server deployment strategies. [5] presents some early results on their DNS anycast stability experiment using a large number of vantage points on the Internet. While this is probably the closest peer related work and our results generally agree, we focus on different anycast deployment strategies, and how they affect the performance of the anycast servers points spread around the Internet.

The effectiveness of anycast in providing redundancy and load sharing has been exploited in a number of proposals. The AS112 project reduces unnecessary load on root nameservers by directing queries for local zones to a distributed black hole implemented via anycast [3]. The use of anycast has also been proposed for finding IPv6 to IPv4 gate-

ways [12] and to implement sink holes for the detection and containment of worm activity [10]. Engel et al. provide results from their measurement of load characteristics on a set of mirror web sites using anycast [8]. Hitesh *et al.* present a scalable design for anycast and use a small subset of the PlanetLab nodes to measure the affinity of existing anycast deployments [4]. While this work has some similarity to ours, their focus is on the design of an anycast scheme. Finally, a number of proposals have looked at alternatives to the existing DNS architecture with the goal of improving query performance [16, 18].

5 Summary

We presented an analysis on the impact of anycast on DNS based on the measurement of four top-level servers. We found that overall, the deployment of anycast is beneficial for the DNS infrastructure since it decreases the average query latency and increases the availability of the DNS servers. However, our study shows that while the number of outages is relatively small, some of them are long in duration ($\approx$ 30% last more than 100 seconds), affected by BGP routing convergence times. We also studied how different deployment strategies play a key role in determining the optimality and robustness of anycast. Finally, we uncovered a trade-off, in which increasing the number of globally visible nodes increases the percentage of queries being directed to the closest cluster, but at the same time de-stabilizes the service offered, in terms of increased server switches and unanswered queries.

While this trade-off is clear from the results presented here, we don't fully understand the underlying mechanisms that connect the scope of BGP advertisements, the rate of flips, and the duration of outages. To do so, would require access to the BGP advertisements at each monitoring point that were unfortunately unavailable.

Acknowledgments

Joe Abley graciously responded to our queries regarding the implementation of anycast in the F-root servers. We would also like to thank Lixia Zhang, Claudiu Danilov and Alexandros Batsakis for their valuable comments that helped us improve this paper.

References

[1] J. Abley. Hierarchical Anycast for Global Service Distribution, 2003. `http://www.isc.org/pubs/tn/?tn=isc-tn-2003-1.html`.

[2] J. Abley. A Software Approach to Distributing Requests for DNS Service Using GNU Zebra, ISC BIND 9, and FreeBSD. In *Proceedings of USENIX 2004 Annual Technical Conference, FREENIX Track*, 2004.

[3] The AS112 Project. http://www.as112.net.

[4] H. Ballani and P. Francis. Towards a deployable IP Anycast Service. In *Proceedings of WORLDS*, Dec. 2004.

[5] P. Boothe and R. Bush. Dns anycast stability: Some early results. Available at http://rip.psg.com/˜randy/050223.anycast-apnic.pdf, 2005.

[6] N. Brownlee and I. Ziedins. Response time distributions for global name servers. In *Proceedings of PAM 2002 Workshop*, Mar. 2002.

[7] R. Elz, R. Bush, S. Bradner, and M. Patton. Selection and Operation of Secondary DNS Servers. July 1997.

[8] R. Engel, V. Peris, and D. Saha. Using IP Anycast for Load distribution and Server Location. In *Proceedings of Global Internet*, Dec. 1998.

[9] L. Gao and F. Wang. The extent of AS path inflation by routing policies. In *Proceedings of Global Internet Symposium, 2002*, 2002.

[10] B. R. Greene and D. Mcpherson. ISP Security: Deploying and Using Sinkholes. http://www.nanog.org/mtg-0306/sink.html.

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

[12] C. Huitema. An Anycast Prefix for 6to4 Relay Routers. *RFC 3068*, June 2001.

[13] RIPE NCC K-Root. http://k.root-servers.org/.

[14] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed internet routing convergence. In *Proceedings of ACM SIGCOMM 2000*, pages 175–187, 2000.

[15] J. Pang, J. Hendricks, A. Akella, S. Seshan, and B. M. an d R. Prisco. Availability, Usage and Deployment Characterisitics of the Domain Na me System. In *Proceedings of the ACM IMC 04*, 2004.

[16] K. Park, V. S. Pai, L. Peterson, and Z. Wang. CoDNS Improving DNS Performance and Reliability via Cooperative Lookups. In *Proceedings of OSDI'04*, Dec. 2004.

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

[18] V. Ramasubramanian and E. G. Sirer. The Design and Implementation of a Next Generation Name Service for the Internet. In *Proceedings of ACM SIGCOMM 2004*, Aug. 2004.

[19] I. Research. Planet Lab. http://www.planet-lab.org/, 2002.

[20] The Route Views Project. Available at http://www.antc.uoregon.edu/route-views/.

[21] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson. The End-to-End Effects of Internet Path Selection. In *Proceedings of SIGCOMM 1999*, Aug. 1999.

[22] N. Spring, R. Mahajan, and T. Anderson. Quantifying the Causes of Path Inflation. In *Proceedings of ACM SIGCOMM*, Aug. 2003.

[23] D. Wessels, M. Fomenkov, N. Brownlee, and K. Claffy. Measurements and Laboratory Simulations of the Upper DNS Hierarchy. In *Proceedings of PAM 2004*, Apr. 2004.