This paper was presented as part of the main technical program at IEEE INFOCOM 2011

Public DNS System and Global Traffic Management

Cheng Huang
Microsoft Research

David A. Maltz
Microsoft Research

Albert Greenberg
Microsoft Research

Jin Li
Microsoft Research

***Abstract*—Cloud service providers operate data centers around the world, and they depend on Global Traffic Management systems to direct requests from clients to the most appropriate data center to serve the requests. While GTM systems have been in-use for years, they are attracting re-newed interests due to the rapid expansion of cloud service providers' networks, the introduction of public DNS systems, as well as new proposals to alter how they should work and what information local DNS servers (LDNS) should make available to drive the GTM systems. This paper uses large-scale measurements conducted from more than 5M clients to establish properties of the current Internet that affect the design of the GTM systems, such as the stretch between a client's actual position and its LDNS from GTM's perspective, the impact of public DNS systems, and the granularity at which GTM decisions should be made. The results can inform the debate over how GTM systems should be designed.**

I. INTRODUCTION

Cloud service providers, such as Amazon, Google, Microsoft and Yahoo, operate data centers around the world, and they depend on Global Traffic Management systems to direct requests from end-users to the most appropriate data center to serve them. Many studies have shown that the end-users are extremely sensitive to response latency and revenue can be significantly impacted because of that [8]. For example, an extra 100 ms delay can decrease revenue by 1% [21]. Hence, directing a request to a sub-optimal data center is undesirable.

Today, the most commonly used mechanism to implement Global Traffic Management leverages the Domain Name System. When a client looks up a FQDN, such as `www.example.com`, the client's host will contact its local domain name server (LDNS) and request the resolution of the FQDN into an IP address. The LDNS will then send a resolution request to the authoritative name server for `example.com`. The authoritative name server uses information available to it — typically just the IP address of the LDNS server — to determine which data center it would like to serve the request. It returns the IP address of the selected data center to the LDNS, which caches the address for the length of time suggested by the authoritative name server. The LDNS then returns the IP address to the client, which finally sends its request to that IP address.

The basic problem to be solved in building a GTM system is coming up with a map $M(\text{client group}) \rightarrow \text{data center}$, where M maps each group of clients to a data center that will serve the clients' requests.[1] For DNS-based GTM systems, as described above, the only possible method of grouping clients today is by their LDNSes — since the only information the authoritative name server has about the clients is the IP address of their LDNS servers, the authoritative name server must assign all the clients behind a particular LDNS to the same data center. In a world where LDNS servers are assigned to clients by the clients' ISPs and the LDNSes are topologically close, this might be a reasonable approximation.

However, there are now multiple *Public DNS systems* being offered by companies like Google, Level3 and OpenDNS [5], [17]. Clients can bypass their ISP-assigned LDNSes and use the Public DNS servers instead, which might be far away from the clients, cause confusion to GTM systems and result in the clients being directed to sub-optimal data centers. To counteract this problem, there are recent proposals that LDNS servers should expose a portion of the clients' IP address to the authoritative name server.

In this paper, we conduct an extensive measurement study to quantify the negative impact on the end-users when using the Public DNS systems as compared to the ISP-assigned LDNS servers, and we evaluate the effectiveness of modifying LDNSes by making client information available to GTM systems. Specifically, we address the following questions:

→ How well do the ISP-assigned LDNS servers approximate the location of the clients from the perspective of GTM systems?

→ How large are the geographic presence of the Public DNS systems, and how do they compare with the LDNS servers deployed by ISPs? How well do the Public DNS servers approximate the clients?

→ If LDNS servers are to expose a portion of the clients' IP address, what granularity of subnet is most appropriate?

Our key findings are:

- ISP-assigned LDNS servers are reasonable approximation for most clients – up to 80% of the clients are within 428 km of their LDNSes. From 17 production data centers distributed over 4 continents, the best data center selected based on the clients' LDNS is only 14 ms further away (from the clients) than the ground-truth optimal, at the 80-percentile.
- The Public DNS systems do *not* have extensive geographic presence. The Google DNS servers are only deployed in 12 locations world wide, while Level3 and OpenDNS in 11 and 10 locations, respectively.
- The Public DNS systems are significantly further away from the clients than the ISP-assigned LDNSes. As a result, the Public DNS systems degrade client-perceived performance. At the 95-percentile, while the best data

[1] In this paper, we assume all data centers are equally capable of serving any client's request, which is typically the common case.

978-1-4244-9921-2/11/$26.00 ©2011 IEEE

center – chosen for those clients using ISP-assigned LDNSes – is already 84 ms further away (from the clients) than the ground-truth optimal, the sub-optimality further degrades to 193 ms, 129 ms and 156 ms, respectively, for those using Google, Level3 and OpenDNS.
- Modifying LDNS servers to pass client information to the authoritative name server can alleviate the adverse impact of the Public DNS systems. Grouping by /24 prefix appears to be a reasonable trade-off between performance improvement and overhead.

II. Methodology

Our analysis is based on data collected through two main mechanisms: a Javascript flight and ping studies. The services with which we work are IPv4-based, so our study does not include IPv6. Our study measures the HTTP GET response time and ICMP ping RTT between clients and 17 production data centers distributed over North America (7), Europe (6), Asia (3), and Australia (1).

A. Flight Experiment

To measure the response time from clients to data centers around the world, we conducted a "flight experiment" [11] in which Javascript code is inserted at end of a commonly accessed commercial webpage for between 1 and 5 percent of visitors. The users receiving the Javascript were selected in an unbiased fashion from among all visitors to the site, which includes significant representation from all areas of the world. We verified that users selected to run our Javascript had the same behavior as other users, so our experiment did not affect their behavior (our Javascript was designed to be completely unobtrusive).

The Javascript begins executing 500 ms after the page load finishes, and this delay is intended to avoid interfering with other objects downloading from the main page. While executing, the Javascript selects the IP address of a data center from a list of 17 IP addresses carried in the Javascript, and then downloads a 10KB image from that data center. The Javascript measures the time to download the image and posts this information back to a logging server along with a GUID created for this execution of the Javascript. The logging server records the GUID, the data center from which the image was fetched, and the IP address of the client. The Javascript executes until the users clicks away from the page or it has probed all 17 data centers. A timeout of 4 seconds was used on each download attempt, and the probing starts at a randomly chosen data centers on each invocation. We chose a 10KB image to download as the data cannot be compressed further, and many important web pages and objects in our organization are roughly this size.

There is no per-user cookie or token used — a NAT or proxy with many users behind it will appear in our logs as a single client IP address associated with many GUIDs (i.e., invocations of the Javascript). No information is available or recorded at to what user action or URL led to the Javascript being downloaded. The client-IP address is the most identifying information, and this same address is available to the service the user made the request of, so our flight does not reveal additional identifying information.

Over 8 weeks, the flight experiment was executed by over 5M unique IP addresses in over 2M unique /24 subnets. The median execution completed probing 5 data centers before the user clicked away from the page.

B. DNS Beacon

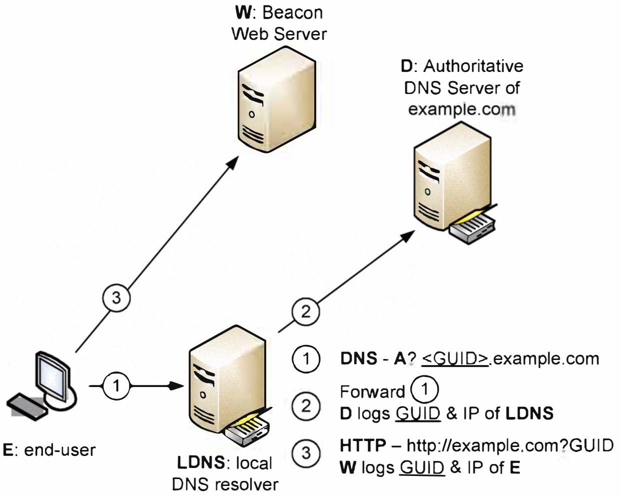


Fig. 1. DNS Beacon

To discover which clients use which LDNS servers, we developed a novel technique called *DNS beacon*. It works as illustrated in Figure 1. The client generates a unique DNS beacon – a hostname with a unique GUID in a domain under our control (say `<GUID>.example.com`). The client sends the hostname to its LDNS for resolution, which then forwards the DNS query to the authoritative name server for `example.com`. The authoritative server records the hostname (thus the GUID) and the LDNS IP. In addition, the client submits the GUID to a web server also under our control in the same domain. The web server records the GUID and the client IP. By joining the logs from both servers on the GUID, the association between client and LDNS is revealed. The DNS beacon is implemented in Javascript incorporated in the flight experiment, and it runs for 5% of the Javascript invocations.

Our DNS beacon found 112,000 LDNS servers, and 3.9M (Client IP address, LDNS) pairs.

C. Direct Ping Measurement

To establish the ground truth of latency, we also deploy active probing agents in each of the data centers. For each collected client IP and LDNS address, the probing agent will send 3 active ICMP Ping requests. For those that do respond, 3 RTT measurements are obtained and the minimum value is used in later analysis. 43% of the client IP addresses were pingable.

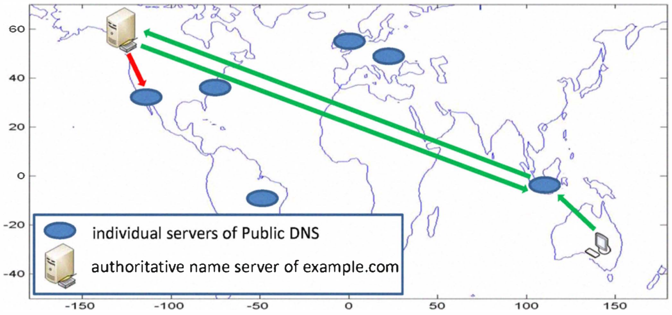

60
40
20
0
-20
-40
-150
-100
-50
0
50
100
150
individual servers of Public DNS
authoritative name server of example.com

Fig. 2. **Illustration of How Public DNS Works.** A Public DNS announces the same *anycast* address from all its servers; a client sends a DNS query to the anycast address; the Public DNS server that receives the DNS query uses a ***unique*** address when contacting the authoritative name server to complete the query (shown in green). The process would fail if the server were to use the anycast address to complete the query (shown in red).

III. Public DNS Systems

Normally, ISPs deploy LDNS servers to provide name resolution service for their clients. The service is typically exclusive, meaning the servers are inaccessible to clients outside of the ISPs. Public DNS providers, on the other hand, deploy publicly accessible DNS servers to provide name resolution service for any clients that choose to use them. Public DNS systems have set up by content providers, ISPs, or third party companies that build a business around the name resolution service. The leading representatives of Public DNS providers of each type are Google, Level3 and OpenDNS [5], [17], respectively.

From perspective of a DNS-based GTM system, when a client chooses to use a Public DNS, it is represented by the corresponding Public DNS servers. The quality of GTM, and hence the client's perceived performance, hinges on the geographic presence of the Public DNS. In this section, we first conduct a measurement study to uncover how many LDNS servers each of the representative Public DNS deploys and where the LDNS servers are located. Given the presence of each Public DNS, we then analyze the client-LDNS association data set extracted from the flight experiment and quantify the popularity of the Public DNS systems; the extent of client-LDNS mismatch; and the latency performance degradation caused by such mismatch.

A. How does Public DNS Work?

We briefly explain the operation of a Public DNS system, using Google Public DNS as an example. As illustrated in Figure 2, Google deploys multiple DNS servers in different cities around the world. All the DNS servers announce the same *anycast* IP address `8.8.8.8`. Say a client from Australia uses Google Public DNS and issues a DNS query to resolve `www.example.com`. The DNS query is sent by the client in a UDP packet towards `8.8.8.8`. BGP and the Internet will route the packet to the anycast-closest Google DNS server, for example, the one in Singapore. The Singapore server then forwards the DNS query to the authoritative name server of domain `example.com`, which we will assume to be in US. The authoritative name server responds to the query and sends the answer to the *source address* of the query.

Note that the source address *cannot* be the anycast address `8.8.8.8`. Otherwise, due to anycast routing, the response will be routed to another Google DNS server in US, rather than back to the Singapore server. Therefore, in order to correctly receive query response, each Google server needs a separate unique IP address. In short, all the Google servers announce the same anycast address to receive DNS queries from clients; each Google server uses a unique address to complete the queries on behalf of the clients. Level3 Public DNS and OpenDNS systems work similarly.

B. Uncovering Public DNS Footprint

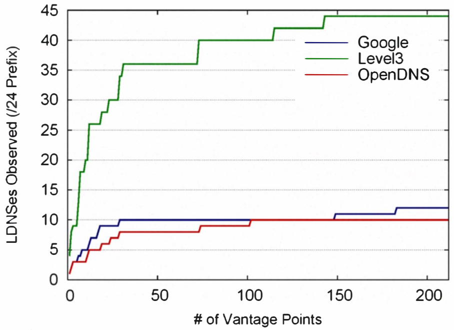


Fig. 3. Public DNS Servers Observed vs. Vantage Points

To uncover the geographic presence of the Public DNS systems, we leverage our DNS beacon technique (Section II-B). We use PlanetLab nodes as clients and configure them to bypass their LDNSes and issue DNS beacon queries directly toward the anycast addresses of each Public DNS system. In this way, our DNS beacon server is able to observe and record the unique IP addresses of the Public DNS servers. By running the experiment from a large number of geographically distributed PlanetLab nodes, we can map the Public DNS systems.

Figure 3 shows that as the number of vantage points increase, the number of discovered Public DNS servers also increase. However, the rate at which new Public LDNS servers are discovered decreases with the number of vantage points, and there are no more servers discovered beyond 180 vantage points. Table I summarizes the number of IP addresses, subnets, cities and continents of the Public DNS systems.

We make the following observations: 1) The *current* geographic presence of the Public DNS systems (10 - 12 cities) are significantly smaller, compared to that of content distribution networks and large content providers which rely on GTM systems. For instance, Akamai deploys caching servers in hundreds of locations worldwide [10]; Limelight has 24 regional distribution centers [14]; Google CDN and Microsoft Edge Network are both deployed in a few tens of locations [4]. Even in North America, where the most Public DNS servers are located, none of the Public DNS system is as widely deployed as the CDNs or the content providers. 2) Level3 and OpenDNS might *not* be targeting their DNS service at the clients outside of North America and Europe, as they do *not* have presence beyond these two two continents. On the contrary, Google *does* have servers deployed in Asia and South America. 3) Google DNS shows one single IP address in each location. The address is likely a virtual IP (VIP), which serves multiple physical servers behind a load balancer. Level3 and OpenDNS are different, as both have more than one IP address in each location. OpenDNS probably deploys one physical server for every IP address, while Level3 appears to be recycling addresses within the same /24 prefix among the servers in each location.

	Google	Level3	OpenDNS

Anycast IP	`8.8.8.8`
	`8.8.4.4`
		`4.2.2.2`
		`4.2.2.1`
			`208.67.220.220`
			`208.67.220.222`
			`208.67.222.220`
			`208.67.222.222`

LDNS IP	12	7758	40
Prefix (/24)	12	44	10
City	12	11	10
Continent	4	2	2
N. America	5	9	8
Europe	4	2	2
S. America	1	0	0
Asia	2	0	0

TABLE I
GEOGRAPHIC PRESENCE OF PUBLIC DNS

C. How Popular are Public DNS Systems?

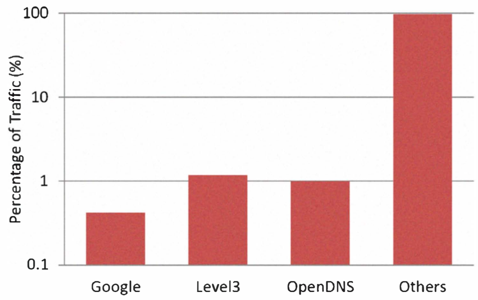


Fig. 4. Popularity of Public DNS

Once the addresses of the individual Public LDNS servers are discovered, it is straightforward to evaluate the popularity of the Public DNS systems. We make the assumption that the population of clients drawn by the web pages carrying our Javascript flight is sufficiently broad that whether a client uses a Public DNS is independent of whether they visit our pages. We can then use the client-LDNS pairs collected from the flight experiment to compute Figure 4, which shows the percentage of clients using Public DNS versus those that do not. In total, about 2.5% of clients are using Public DNS. In Internet scale, this is a quite significant number. Google Public DNS is the least popular one (about half of the other two). But, considering the fact that it was released only 6 months ago (from the time of our experiments), the popularity is quite impressive.

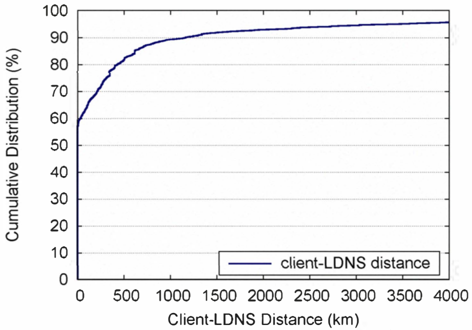


Fig. 5. Client-LDNS Distance (global)

IV. PROBLEM OF CLIENT-LDNS MISMATCH

DNS-based GTMs use LDNS as an approximation of its associated clients. The quality of the GTMs and, in turn, the performance perceived by the clients are affected by the proximity of the clients with their LDNS. Earlier studies concluded that *client-LDNS mismatch* is a severe problem. For instance, Shaikh et al. [20] showed that the Internet paths to clients and their LDNS servers were typically disjoint and could differ 8 hops or more. Mao et al. [15] showed that only 64% of the clients shared the same AS-cluster with their LDNSes, and the ratio dropped to 16% in network-aware clustering.

A key question left unanswered by those studies is how such client-LDNS mismatch translates into performance degradation when the clients access cloud services with globally distributed data centers. In addition, over the decade since those studies were completed, major trends are significantly re-shaping the Internet architecture. One of the most notable ones is that large content distribution networks and content providers are aggressively building their own backbone infrastructure to largely bypass ISP networks and reach end-users directly. In the following sections, we will re-visit the client-LDNS mismatch problem under the current Internet environment. Moreover, we will answer the ultimate question how the mismatch affects the accuracy of a GTM solution.

A. How Far are Clients from LDNSes?

Using the client-LDNS pairs collected from the flight experiment, we first examine the geographic distance between the clients and their LDNSes. The Quova IP GeoLocation database is used for this purpose.

1) A Global View: Figure 5 shows the cumulative distribution of the geographical distance between each client-LDNS pair. We observe that most clients are in fact *not* far from their LDNSes. It is clear that the client-LDNS distance is within 27 km for 60% of the clients and less than 428 km for 80%. When mapped to network latency, such distances correspond to small values. This explains why almost all GTM solutions used in current production systems are DNS-based [10], [13]. On the other hand, a non-trivial percentage of clients are *significantly* distant from their LDNSes – the client-LDNS distance is more than 3,393 km for 5% of the clients.

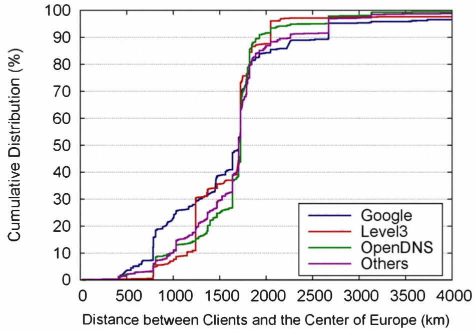


Fig. 6. Distribution of distance between European clients and the geographic center of Europe.

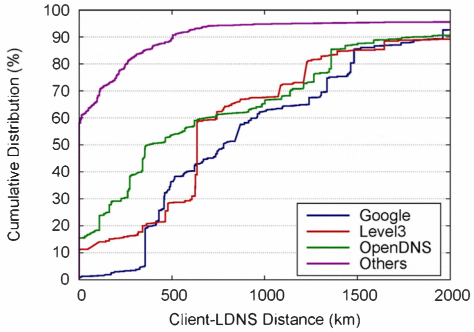


Fig. 7. Distance between European clients and their LDNS servers.

Figure 5 shows the results for all the client-LDNS pairs. Next, we separate out the clients not using Public DNS and compare them to those that do. Also, as the Public DNS systems do *not* have presence on every continent, the comparison is more meaningful and informative if the results are broken down by continent.

2) Breakdown View – Europe: We first analyze the clients located in Europe. One might wonder whether there is inherent bias due to the client population itself. For instance, a client switches from its ISP-assigned LDNS to Public DNS *only when* the ISP-assigned LDNS is far away (and thus causing poor performance). As a simple sanity check, we take the geographic center of Europe as a reference point and calculate the distance from each client to the center. Figure 6 plots the cumulative distribution of distance for each DNS system. From the distributions, there appears to be no bias among the client population — use of Google, Level3, or OpenDNS is independent of location.

Comparing Figure 7 with Figure 5, we observe the client-LDNS distance is much smaller in Europe than over the entire world. In addition, we observe that the difference between ISP-assigned LDNS and Public DNS is remarkable. At the 80-percentile, the client-LDNS distance is 253 km, while the client-Public DNS distance is 1,464 km, 1,228 km and 1,358 km for Google, Level3 and OpenDNS, respectively. Interestingly, despite being deployed in more locations (4 vs. 2), the Google Public DNS appears to be further away from the clients than Level3 and OpenDNS.

3) Breakdown View – North America: Next, we examine the clients located in North America. Figure 8 shows that, compared to Europe, the client-LDNS distance is significantly larger on this continent. At the 80-percentile, the distance is 616 km for the clients using ISP-assigned LDNS, while it is 2,206 km, 2,059 km, 1,504 km for those using Google, Level3 and OpenDNS, respectively. The fact that Google Public DNS is deployed in fewer locations in North America (5 vs. 9 and 8, respectively) likely causes those clients using Google to be be even further away from their DNS servers than those using Level3 and OpenDNS.

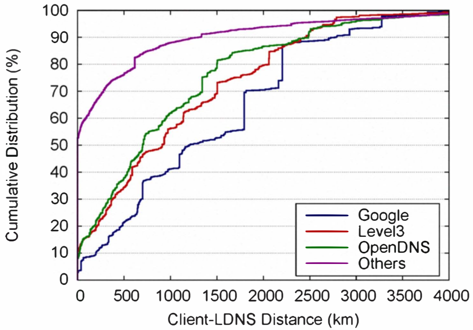


Fig. 8. Client-LDNS Distance (N. America)

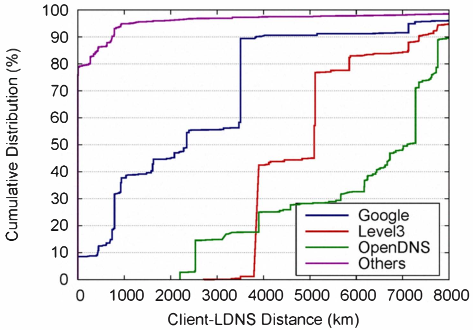


Fig. 9. Client-LDNS Distance (S. America)

4) Breakdown View – South America: South America is interesting to examine as it is one of the continents where Google Public DNS has presence, while Level3 and OpenDNS do *not*. Figure 9 shows that the clients using Google are far closer to their LDNSes than those using Level3 and OpenDNS. To further examine the benefit of having presence on the continent, Figure 10 plots, for all the clients in South America, the percentage of their LDNSes on different continents.

We observe that, for those South American clients using

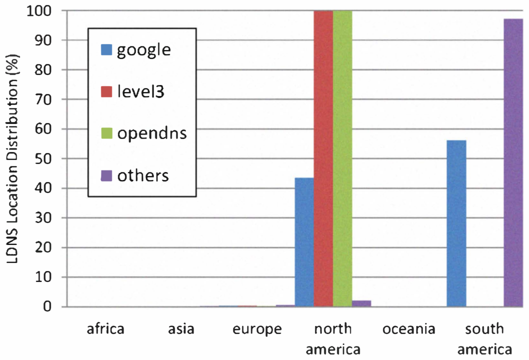


Fig. 10. Percentage of LDNSes on each continent (S. American clients).

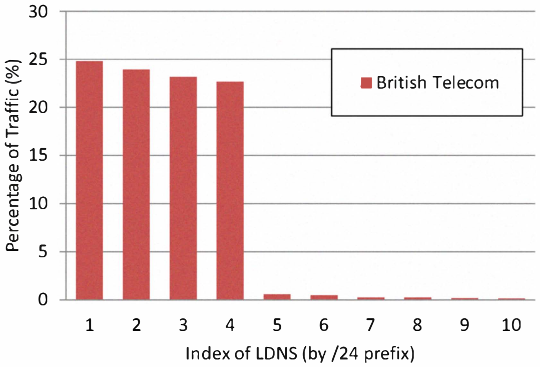


Fig. 11. LDNS used by British Telecom clients

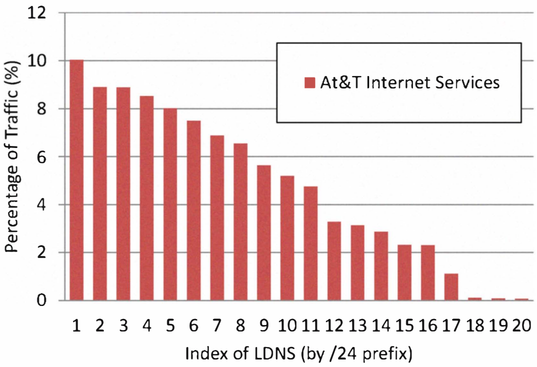


Fig. 12. LDNS used by AT&T clients

	British Tel.	AT&T Inet.
Total Clients	81823	58829
Total LDNS	754	1192
Total Subnets (/24)	377	679

TABLE II
LDNS USED BY CLIENTS FROM NATIONAL ISPS

	British Tel.	AT&T Inet.
LDNS IP Prefix	4	17
# of Cities	3	17

TABLE III
LDNS TRULY DEPLOYED BY ISPS

Level3 and OpenDNS, their associated DNS servers are mostly in North America. This is understantable, as Level3 and OpenDNS do *not* have presence in South America. On the other hand, more than 56% of the clients using Google are associated with DNS servers on the same continent. Clearly, these clients are much closer to their DNS servers. For the rest (about 44%) clients, they are still associate with DNS servers in North America. The fact, that anycast routes the DNS queries from those clients to North America, suggests that the peering relationship between Google and South America ISPs has a lot of room to improve. Again, we observe that ISP-assigned LDNS servers are much closer and more than 97% locate on the same continent as the clients.

B. ISP-assigned LDNS vs. Public DNS

The evidence so far has clearly shown that the client-LDNS distance is smaller when the clients use LDNS provided by their ISPs, rather than by the Public DNS systems. This is intuitive for regional ISPs. However, for national ISPs, the client-LDNS distance depends on how many LDNS servers and where they are deployed by each ISP. Knowing the association of LDNS to ISP allows us to compare the footprint of ISP-assigned LDNS against that of Public DNS. This is important to GTM, as the smaller footprint will be the dominating factor in determining GTM performance.

So far, we have been using the term “ISP-assigned LDNS” loosely. In fact, all LDNSes are *not* assigned by ISPs. For example, an organization can run its own DNS server and the clients in the organization can use the server as their LDNS. Obviously, this LDNS is no longer assigned by the client’s ISP. Here, we are interested in discovering the LDNSes that are truly deployed by the ISPs. Therefore, a straightforward approach, which simply counts all the LDNS servers belonging to the ISPs’ netblocks, does *not* work. As illustrated in Table II, we examine British Telecom and AT&T Internet Services, two top ISPs from Europe and North America. There are hundreds of LDNS servers used by the clients from the two ISPs. Even counting subnets, the quantity is still very large. It is unlikely that all the hundreds of DNS servers are deployed by the two ISPs.

To distinguish truly ISP-assigned LDNS, we remark that LDNSes operated by organizations are like to serve only a small number of clients and remain *unpopular*. On the contrary, LDNS servers deployed by ISPs will serve many more clients. To validate this intuition, we group the LDNS servers by /24 prefix and rank the groups by their popularity. As shown in Figures 11 and 12, there is a clear separation between popular and unpopular LDNS subnets. Hence, this method allows us to determine that British Telecom deploys LDNSes in 4 subnets while AT&T Internet Services in 17 subnets. Using the GeoLocation database, we discover that BT’s LDNS servers are in 3 cities and AT&T’s are in 17 cities (Table III). Compared to the Public DNS systems, the geographic presence of major ISPs are much bigger. This explains why the clients are closer to their ISP-assigned LDNSes than to the Public DNS systems.

V. LATENCY PERFORMANCE

In this section, we investigate how client-LDNS mismatch affects client perceived latency performance.

A. LDNS as an Approximation

Recall that for every client and LDNS address collected through the flight experiment, the direct probing agent running in each data center sent Ping probes to it. For those responded to Ping, the minimum value of three RTT measurements is taken as the ground truth latency. The data center with the shortest latency can then be determined for each client (called *client-best*), and the latency between the client and the client-best data center is the best latency performance the client can achieve. Since DNS-based GTMs use the client’s LDNS to make the decision, they will choose the data center with

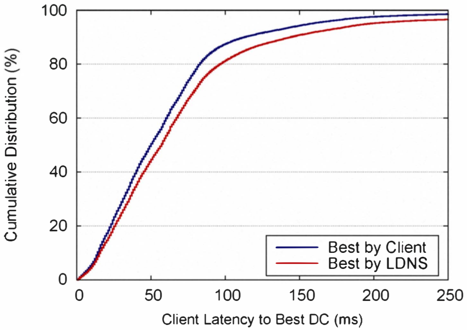


Fig. 13. LDNS as an Approximation

the shortest latency to the LDNS (called *LDNS-best*). The latency between the client the LDNS-best data center is what is achieved by GTMs. The latency degradation is the gap between the client-best and LDNS-best distributions, as shown in Figure 13.

We can make the following observations: 1) the two distributions are fairly close, which means that for a large percentage of clients, the difference between between client-best and LDNS-best is quite small. This again confirms that LDNS is a reasonable approximation of client; 2) at high percentile, however, the distance between the two distributions is non-negligible. For instance, at the 95 percentile, the client-best latency is 157 ms, while the LDNS-best latency is 197 ms. Once amplified through applications, 40 ms difference *per round trip* does indeed degrade client perceived performance.

B. Impact of Public DNS Systems – an Overview

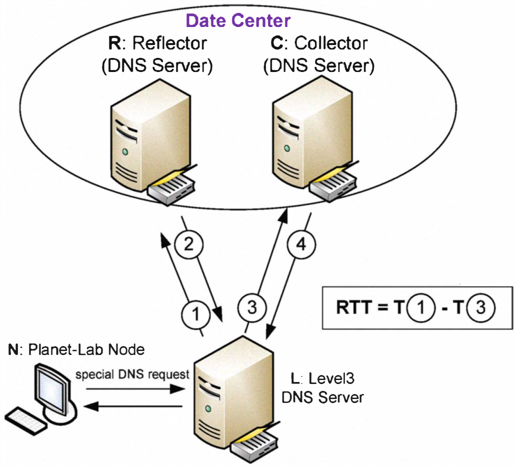


Fig. 14. Determine LDNS-best for Level3

For the clients using Public DNS, to find their corresponding LDNS-best data centers, we need to know latency from each data center to all the Public DNS servers. The DNS servers of Google and OpenDNS respond to Ping, so it is straightforward to obtain latency. Level3 DNS servers, however, do *not* respond to Ping and we need another way to measure latency. To this end, we leverage a previously developed technique – *DNS reflection* – to obtain such measurement. On a high level, a DNS query of a special hostname is sent to a Level3 DNS server. The hostname belongs to a domain whose authoritative name server locates in a target data center. The authoritative name server responds to the DNS query in such a way that the Level3 DNS server is triggered to immediately send another DNS query to a secondary DNS server in the same data center. As illustrated in Figure 14, by examining the time difference between receiving the first and the second DNS query at the data center, the latency between the data center and the Level3 DNS server can be readily determined. For details, please refer to [9].

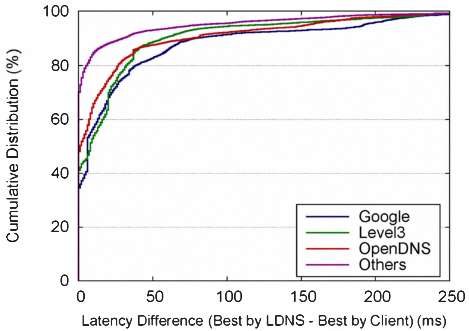


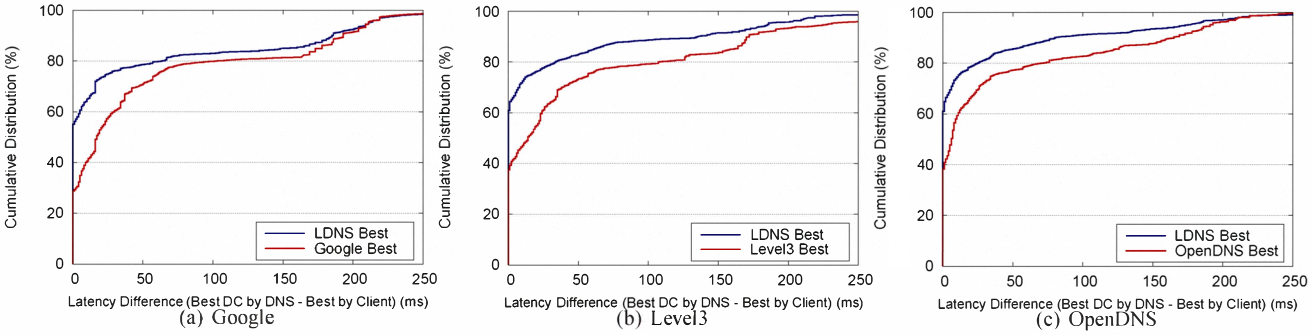
Fig. 15. Impact of PubDNS

Now, we can determine the LDNS-best data centers for the clients using Google, Level3 and OpenDNS. Figure 15 shows the breakdown distributions of the latency difference between the client-best and the LDNS-best data center. We observe that the latency difference is very small for a large percentage of the clients. For instance, at the 90 percentile, the difference is 31 ms for the clients using ISP-assigned LDNS, compared to 81 ms using Google, 58 ms using Level3 and 78 ms using OpenDNS. However, at the 95 percentile, the difference quickly increases to 84 ms for the clients using ISP-assigned LDNS, compared to 193 ms using Google, 129 ms using Level3 and 156 ms using OpenDNS. Hence, the performance degradation due to Public DNS is substantial for a non-trivial percentage of the clients.

C. Impact of Public DNS – a Detailed Comparison

While the latency difference comparison in the previous section is informative, it is *not* an apple-to-apple comparison, as the client populations are not exactly the same across all the distributions. A more fair comparison is to analyze the latency performance for exactly the same clients. To this end, we group the clients by their /24 prefix. For a large number of prefixes, we observe that there are a few clients using the Public DNS systems, while most others use ISP-assigned LDNS. For the clients using Public DNS, we record their associated DNS servers. We assume these DNS servers will be used by the other clients in the same prefix, if they were to switch from ISP-assigned LDNS to Public DNS. Because such grouping resulted in difference prefixes for different Public DNS, we present the comparison separately for each Public DNS, as shown in Figure 16.

Using Google as an example (Level3 and OpenDNS are similar), Figure 16(a) shows the distributions of latency difference between client-best and DNS-best, when the clients



(a) Google (b) Level3 (c) OpenDNS

Fig. 16. Impact of PubDNS (breakdown)

use ISP-assigned LDNS or Public DNS, respectively. The difference between client-best and DNS-best is the performance degradation caused by client-LDNS mismatch. If a GTM solution were able to determine the client-best data center for a client, then it can eliminate such performance degradation.

It appears that Google, as an cloud service provider, exactly achieves this goal by offering its own Public DNS service. When clients switching from ISP-assigned LDNS to Google Public DNS, their performance accessing Google services will improve, as Google's GTM can now observe the clients' IP and select data centers that are client-best, rather than DNS-best. However, when the clients access any other cloud services, their performance will inevitably degrade. The best data center determined by the GTMs of those services, can only be DNS-best with respect to Google DNS servers. Because the Google DNS servers are further away from the clients than ISP-assigned LDNS, the performance perceived by the clients will be worse than before switching to the Public DNS system.[2]

VI. Granularity of Client IP Based Maps

Partly in response to criticism that public DNS systems reduce the performance of clients that use them, at least when the clients access services other than those operated by the public DNS provider, the IETF is considering a proposal [2] to change LDNS servers to pass to the authoritative some information about the client IP address along with the FQDN to be resolved, an idea also explored by Shaikh et al. [20] back in 2001.

This, in turn, raises the question of how many bits of the client IP address should be forwarded. That is, if client IP addresses will be grouped by subnet, what netmask length should be used? The current proposal suggests passing on the top 24-bits (i.e., the /24 subnet), but either more or fewer are possible.

Since the LDNS will cache answers from the authoritative name server for each subnet, as netmask length increases there will be more subnets, more cache misses, and the LDNS will make more requests to the authoritative name server. Owners of authoritative name servers have raised the concern their servers might not be able to handle this additional load. On the other hand, if the netmask is too short, the GTM solution will suffer and again users will see worse latencies than they should.

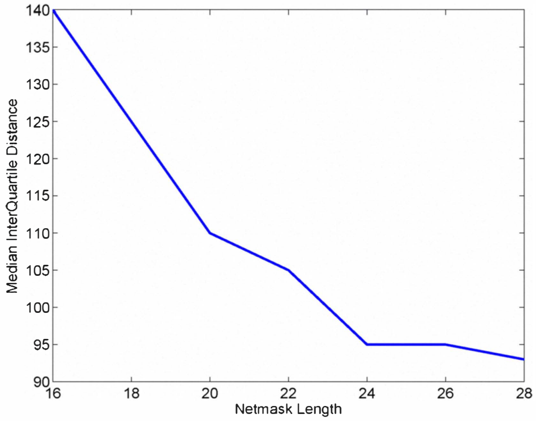


Fig. 17. Width, as measured by IQD, of the response time distributions falls as netmask length increases. Diminishing returns set in at a 24-bit subnet.

A. Evaluation Metric

One approach would be to cluster users into the maximally sized subnets that all have the same best data center, and have the authoritative name server return the netmask length of this subnet along with the IP address in response to a resolution request. Here, we start with the approach of finding the best subnet size across all users, and the impact of using larger or smaller subnets.

We take the response time data from the Javascript flight and group it into subnets of with netmask length k by the client IP address. For each client subnet, we then compute the response time distribution to each data center. We then compute the "width" of each distribution using inter-quartile distance (IQD) of the distribution (i.e., the 75th percentile minus the 25th percentile). This produces for each netmask length k a distribution of IQDs — the intuition behind the methodology is that as clients that are topographically dispersed are grouped together in the same subnet, the width of the response time distribution to each data center will increase.

B. Impact of Netmask Length

Figure 17 shows the median of the distribution of interquartile distances as the netmask length used by LDNS servers increases from 16-bits to 28-bits. As expected, the widths of the response time distributions fall as the netmask gets longer,

[2] Of course, the performance will be even more sub-optimal than from a client-best data center.

since clients with IP addresses closer together are more likely to be topologically closer to each other and hence have similar response times to each data center.

Diminishing returns appear to set in at 24-bits, so the quality of the GTM solution will not be improved if LDNS servers pass on more than 24-bits of the client IP address to the authoritative name servers. If load on the name servers is an issue, the load could be cut by 16x by decreasing the netmask length from 24 to 20 bits, at a cost of increasing the median width of the response time distributions by 15 ms.

VII. Related Work

This paper updates the work of Shaikh et al. in 2001 [20] and Mao et al. in 2002 [15] to measure the proxmity of web clients and LDNS servers. In those earlier studies, proximity is quantified using an extensive set of metrics, including AS, network-aware cluster, Internet path hop count, etc. However, there was one important question left unanswered, that is, how the proximity ultimately affects client perceived performance when service is delivered from many geographically distribued data centers. In this paper, we not only examine new issues that did not exist at that time, such as the effect of public DNS systems on the client-LDNS relationship, but also answer the ultimate question how client-LDNS mismatch impacts global traffic management and affects client response time distribution. We also look at additional issues, such as the granularity of client IP subnets, and we introduce new techniques for associating clients with their LDNS and discovering LDNS deployed by national ISPs.

Many alternative systems for Global Traffic Management have been proposed, including light-weight probing, anycast, combinations of anycast with probing, and entirely new, non-DNS based systems that use new code running on the clients to resolve a service name to an IP address [1], [3], [7]. DNS-based GTM remains extremely popular among major service providers, so our work focuses on DNS-based systems. Note although DNS-based GTMs are subject to client-LDNS mismatch, they can be enhanced through URL-rewriting [12], which is a late-binding process beyond the DNS resolution stage. Such technique redirects subsequent client requests to desirable data center choices, even if the initial one based on LDNS is *not* optimal.

A significant body of prior work looks at mapping the location of clients using geographic or virtual coordinate systems, often using probing or data mining of user provided information to create the maps (a few of the early efforts among these many papers are [16], [18]). Our work differs in that we work directly with measurements of latency distribution, without converting to an intermediate coordinate system, and we evaluate the implications for DNS-based GTM systems.

The DNS reflection technique used to measure the latency between Level3 DNS servers and the data centers bears similarity to the widely adopted King technique [6]. The gist of both approaches is to use modified DNS query and response traffic to measure Internet latency.

VIII. Summary

In summary, through a large-scale measurement study, we observe that using ISPs' LDNS servers to approximate the location of the clients behind them results in a reasonable data center selection for a large number of clients.

When a client bypasses its ISP's LDNS and uses a public DNS system, GTM systems are more likely to redirect the client to a sub-optimal data center. This effect has been reported anecdotally in blogs [19], and we have confirmed it quantitatively.

Using our measurement study, we also evaluate recent proposals on altering how DNS systems work and what information local DNS resolvers should make available to drive the GTM systems. We hope the results will shed light on the debate over how GTM systems should be designed.

Acknowledgment

We thank Nick Holt, Zee Xu, Gregory Vandenbrouck, Eric Schurman, Allen Callaway and Jean-Francois Peyroux at Microsoft Corporation for their help with our experiments.

References

[1] R. L. Carter and M. E. Crovella. Server Selection using Dynamic Path Characterization in Wide-Area Networks. In *INFOCOM*, 1997.

[2] C. Contavalli, W. van der Gaast, S. Leach, and D. Rodden. Client ip information in DNS requests. IETF Internet Draft draft-vandergaast-edns-client-ip-00.txt, Jan 2010.

[3] M. J. Freedman, K. Lakshminarayanan, and D. Mazieres. Oasis: Anycast for any service. In *NSDI*, 2006.

[4] Google Data Center FAQ. http://www.datacenterknowledge.com/archives/2008/03/27/google-data-center-faq/.

[5] Google Public DNS. http://code.google.com/speed/public-dns/.

[6] K. P. Gummadi, S. Saroiu, , and S. D. Gribble. King: Estimating Latency between Arbitrary Internet End Hosts. In *IMW*, 2002.

[7] J. Guyton and M. Schwartz. Locating nearby copies of replicated Internet servers. In *SIGCOMM*, 1995.

[8] J. Hamilton. The cost of latency. http://perspectives.mvdirona.com/2009/10/31/TheCostOfLatency.aspx, October 2009.

[9] C. Huang, N. Holt, Y. A. Wang, A. Greenberg, J. Li, and K. W. Ross. A DNS Reflection Method for Global Traffic Management. In *USENIX ATC*, 2010.

[10] C. Huang, Y. A. Wang, J. Li, and K. W. Ross. Measuring and Evaluating Large-Scale CDNs. In *Microsoft Research Technical Report*, 2008.

[11] R. Kohavi et al. Practical Guide to Controlled Experiments on the Web: Listen to Your Customers not to the HiPPO. *KDD*, 2007.

[12] B. Krishnamurthy, C. Wills, and Y. Zhang. On the Use and Performance of Content Distribution Networks. In *IMW*, 2001.

[13] R. Krishnan, H. V. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao. Moving Beyond End-to-End Path Information to Optimize CDN Performance. In *IMC*, 2009.

[14] Limelight Global Network. http://www.limelightnetworks.com/platform/cdn/.

[15] Z. M. Mao, C. Cranor, F. Douglis, M. Rabinovich, O. Spatscheck, and J. Wang. A precise and efficient evaluation of the proximity between web clients and their local dns servers. In *USENIX*, June 2002.

[16] T. S. E. Ng and H. Zhang. Predicting internet network distance with coordinates-based approaches. In *INFOCOM*, 2002.

[17] OpenDNS. http://www.opendns.com/.

[18] V. N. Padmanabhan and L. Subramanian. An investigation of geographic mapping techniques for internet hosts. In *SIGCOMM*, 2001.

[19] D. Rayburn. Google's New DNS Service Has One Major Flaw, Poor Performance. http://blog.streamingmedia.com.

[20] A. Shaikh, R. Tewari, and M. Agrawal. On the Effectiveness of DNS-based Server Selection. In *INFOCOM*, 2001.

[21] S. Souder. High performance web sites: 14 rules for faster loading pages. http://stevesouders.com/docs/velocity-20090622.ppt - Statistic attributed to Greg Linden., June 2009.